****

**Princess Sumaya University for Technology**

**Department of Data Science**

**Information Systems Security Course**

**Phishing URL Synthesis and Detection**

Done by:

Rama AlAlawneh 20190153

Sara Abbas 20190716

1. Abstract

Phishing detection solutions have long proved effective in avoiding phishing attacks, a machine learning approach can be deployed for the prevention of such attacks before they cause any damage. This paper introduces a URL-based NLP focused machine learning approach to detect whether a website URL is phishy or legitimate. We gathered data from a variety of up to date sources, so we can capture the newly evolving phishing tactic patterns. We have extracted several features from the lexical characteristics of around 340,000 URLs and trained a decision tree model to classify them, achieving an accuracy of 92.54%.

1. Introduction

Phishing is one the most easily deployed yet catastrophic security threats to web clients or even specialist organizations. In a phishing assault, the assailant utilizes faked messages and sites to attempt to get client-delicate confidential information, or install malicious software on the victim’s device. Phishing takes advantage of people’s vulnerabilities of trusting any source that looks credible. Falling into a phishing trap is highly reliable on human error, this is why various security methods have been developed to detect phishing attacks before they can cause any damage to the end user.

Phishing detection solutions can be divided into two categories, content-based, and URL-based. Our solution is a URL-based one as we study features that can be acquired from the website URL itself regardless of its contents. This is a safer approach to detect phishing attacks, as the detection can occur before interacting with the website to study its contents.

This paper introduces an NLP focused machine learning solution to detect phishing URLs that can be implemented in services that are honeypots for cryptanalysts to lure their phishing victims.

2.1 Dataset

One of the challenges faced in looking at datasets for phishing URLs is that there are usually more benign URLs collected than malicious ones. This creates an imbalance in the data which could cause some bias towards the classification of real URLs in contrast with fake ones. This is shown in one of the datasets *[7]* we have used in our study with around 390,000 records of legitimate URLs and 150,000 of phishing URLs. We have only decided to use a part of this dataset to increase our data records.

We have gathered URLs from various datasets and sources [4] *[7]* *[9]*and merged them into one balanced dataset. Phishtank was one of the data sources that we extracted purely phishing URLs from to fill the gap between the two classes of data *[9]*. The reason we used Phishtank is that it contains the latest phishing URLs that web users encounter, this could help our model pick up the latest patterns used by cryptanalysts to lure their phishing victims.

We ended up with a dataset of 338,922 URLs of which almost 50% are phishing URLs and 50% are legitimate ones. This initial dataset consisted of two features; the full URL and its label- whether it’s a phishing URL or a legitimate one.

*Figure 2.1 Frequency of dataset Target column classes*

4. Literature review

To solve the problem of imbalanced data, some studies have implemented deep learning with Generative Adversarial Networks (GAN) to generate synthetic fake URLs, then use them as part of the training dataset *[8]* [6]. GANs train two models concurrently, a generative model which generates synthetic data, and a discriminative model which classifies the now balanced data. Other researches have used different measures to oversample the data.

As for prediction and modeling, recent studies have conducted systematic reviews of which models were best suited for phishing URL classification, Decision Trees were found to be the best, followed by Random Forest, SVM, and Logistic Regression *[12]*.

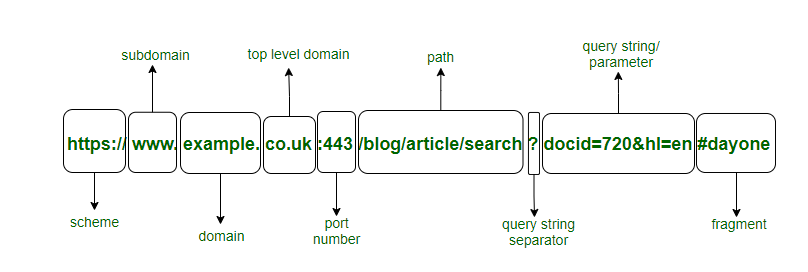
Table 1: Other researches models used and accuracy

|  |  |  |
| --- | --- | --- |
| Paper | Models used | Accuracy |
| Lightweight URL-based phishing detection using natural language processing transformers for mobile devices[5] | ANN | 86.2% |
| Semi-supervised Conditional GAN for Simultaneous Generation and Detection of Phishing URLs [6] | GAN | 95.52% |
| Detection of Phishing Websites using an Efficient Machine Learning Framework[10] | Random Forest | 91.4% |
| Detection of Phishing URL using Ensemble Learning Techniques[3] | Random Forest | 96.15 % |
| How to Detect Phishing Website Using Three-Model Ensemble Classification[11] | Random Forest, Support Vector Machine, Decision Tree | 98.52% |

1. Methodology

3.1 URL structure

A URL (Uniform Resource Locator) simply locates a certain internet resource and is used to reference and retrieve it. A URL can be dissected into three or four components according to IBM *[2]*; the scheme, hostname or domain, path, and query if specified. Every URL starts with a scheme that specifies the network protocol, usually https or http. The main difference between these two protocols is that https is more secure, as it uses encryption for the transfer of data packets. A URL’s domain is the string identifier used by a domain name system (DNS) to determine the online resource, this is easier to type out and remember than the ip address of the online resource. The path represents the route to a specific resource within the domain that a web user wants to reach.



*Figure 3.1 URL main and sub components*

3.2 Feature Extraction

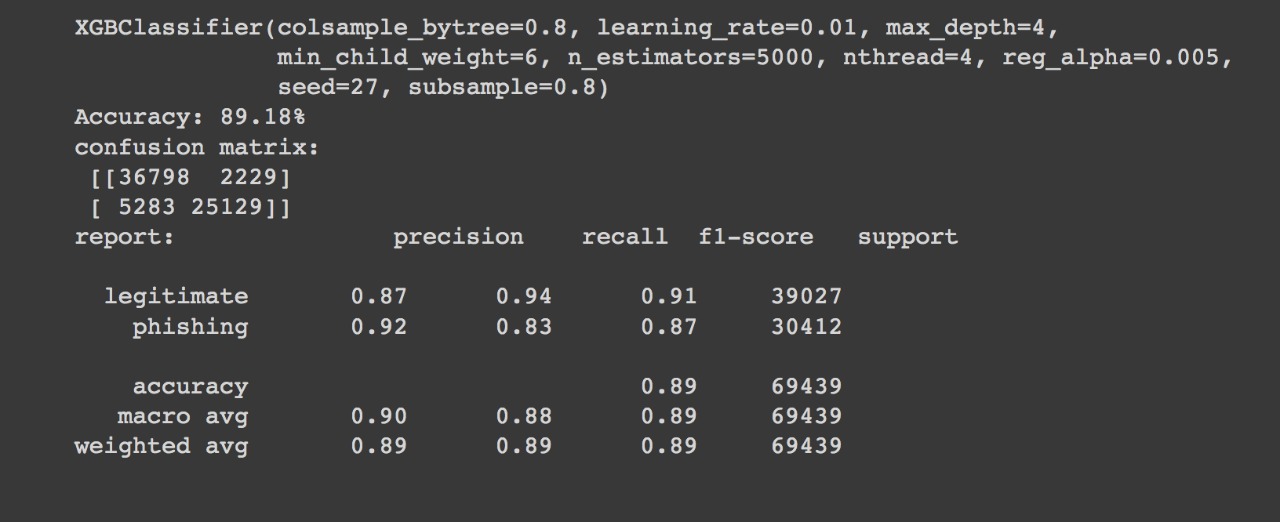
Our approach of dealing with URLs for classification was a Natural Language Processing (NLP) focused one. We extracted lexical features from the URLs relating to each’s components to train our model.

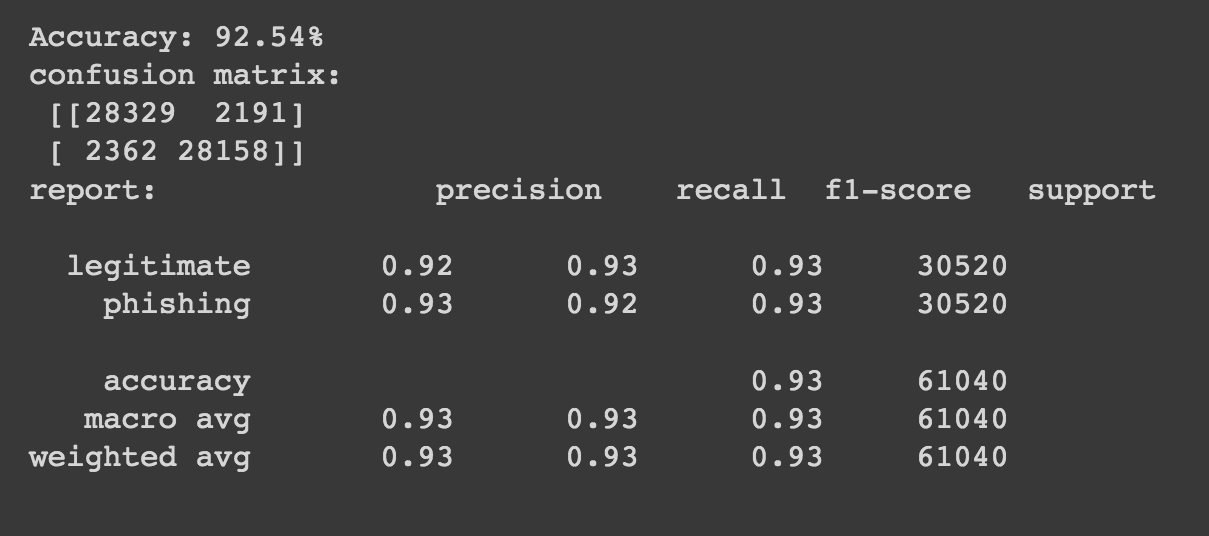
Two important libraries we’ve used in our solution were urllib and re libraries. Urllib is a python package that enables us to parse and work with different components of the url and includes some functions that aid in dealing with URLs. Re is the regular expression library that is commonly used in NLP applications for matching operations.

* ‘text\_tokenized’: Essentially, tokenization is a natural language processing technique that breaks down a phrase, sentence, paragraph, or entire text document into smaller units, such as individual words or terms. Each of these smaller units is called a token, so we broke the URL into tokens.
* ‘text\_stemmed’: Stemming is a natural language processing technique that reduces inflections of words to their basic form, so we stemmed the tokens that we created in the previous ‘Text\_tokenized’ feature.
* ‘text\_sent’ : In this feature, we joined all the stemmed words together and converted them into strings instead of a list.
* ‘URL\_length’ is a self-explanatory feature, it denotes the length of the full URL. We chose this feature out of the assumption that sometimes phishing attackers use long URLs to hide the fake forged text characteristics within the long string.
* We represented the domain of the URL as a single string feature ‘domain’ which could actually be used for further feature engineering. To draw out the domain in a URL, we parsed it then used the scheme out of the 6 component tuple from the urllib.parse module.
* ‘has\_ip’ feature uses the regular expression library to search for the two versions of ip addresses IPv4 and IPv6 in both binary and hexadecimal representation within the URL. It is sometimes an indicator when the domain name in the URL is an ip address that the URL is a phishing website as it’s not common practice to reference a web resource using its ip address.
* The ‘scheme\_https’ feature is an indicator of whether the URL’s protocol is an https (secure) protocol or not. To implement this feature, we extracted the scheme of the url after parsing it. If the scheme is ‘https’, then this feature’s value is 1, else 0. Contrary to our hypothesis, this feature wasn’t a strong indicator taken into consideration that a recent phishing activity trend report in 2020 [1] indicated that 74% of all phishing URLs start with ‘https’, so this feature might not be a strong indicator for the classification of whether the URL is a phishing or legitimate one.

3.3 Modeling

The problem at hand is a binary classification problem, at first we tried using the xgboost model which is a decision tree-based ensemble machine learning algorithm that uses gradient boosting frameworks. It gave us an accuracy of 89.18%.

*Figure 3.2.a Accuracy, confusion matrix and classification report of XGBoost model*

The other model we tried working on is decision trees, A decision tree is a type of supervised machine learning used to classify or make predictions based on how a previous set of questions was answered, it gave us an accuracy of 92.54% without using any hyperparameters.

*Figure 3.2.b Accuracy, confusion matrix and classification report of Decision trees model*

Taking into consideration that our model performed really well with the default parameters for decision trees, we tried different means for tuning our parameters but decided to stick with our original accuracy.

1. Machine learning approach compared to traditional Phishing URL detection

Standard ways of identifying malicious URLs are like blacklisting which are matching URLs to repositories of known malicious ones, as well as heuristic base lists which are lists of signatures of common attacks [13]. The main problem of these traditional methods is the latency in storage and updating of all newly evolving and generated phishing URLs, so a machine learning approach which can predict whether the URL at hand is a phishing URL regardless if it has been already deployed or not would be a much sufficient approach than lookups in existing datasets.

1. Findings and conclusions

For the two different models we used, we can see that decision trees achieved a higher accuracy (92.54%) than the xgboost model (89.18%), so in our case we would rather use decision trees for the classification of phishing URLs since it is more accurate.

For future work, we plan to validate our results even further by experimenting with other models, datasets, and implementing more feature engineering schemes. We’re also aiming to take a more deep learning oriented approach, as most progressing researches in the field of phishing detection are utilizing Generative adversarial networks for synthesizing new phishing URLs and using them for training their network.

1. References

*[1] APWG | Phishing Activity Trends Reports*. (2022). PHISHING ACTIVITY TRENDS REPORTS. <https://apwg.org/trendsreports/>

*[2] The components of a URL*. (n.d.). © Copyright IBM Corporation 2019. <https://www.ibm.com/docs/en/cics-ts/5.1?topic=concepts-components-url>

*[3] Detection of Phishing URL using Ensemble Learning Techniques*. (2020). Detection of Phishing URL Using Ensemble Learning Techniques. <http://norma.ncirl.ie/4509/1/sharadrajendraparmar.pdf>

[4] Hannousse, A. (2021, June 25). *Web page phishing detection*. Mendeley Data. <https://data.mendeley.com/datasets/c2gw7fy2j4/3>

*[5] ScienceDirect. (n.d.). Cloudflare. (2020, June). Detection of Phishing Websites Using an Efficient Machine Learning Framework.* [*https://www.sciencedirect.com/science/article/pii/S1877050921014368*](https://www.sciencedirect.com/science/article/pii/S1877050921014368)

[6] Kamran, S. A. (2021, August 4). *Semi-supervised Conditional GAN for Simultaneous Generation and. . .* arXiv.Org. <https://arxiv.org/abs/2108.01852>

*[7] Phishing Site URLs*. (2020, July 21). Kaggle. <https://www.kaggle.com/datasets/taruntiwarihp/phishing-site-urls>

*[8] Phishing URL Detection with Oversampling based on Text Generative Adversarial Networks*. (2018, December 1). IEEE Conference Publication | IEEE Xplore. <https://ieeexplore.ieee.org/document/8622547>

*[9] Phishtank*. (2022). [Dataset]. Retrieved 10 June 2022, from <http://data.phishtank.com/data/online-valid.csv>.

*[10] ResearchGate*. (n.d.). Lightweight URL-Based Phishing Detection Using Natural Language Processing Transformers for Mobile Devices. <https://www.sciencedirect.com/science/article/pii/S1877050921014368>

*[11] A Thesis Submitted in Partial Fulfilment of the Requirements of the Master Degree in Computer Science*. (2020, June). A Thesis Submitted in Partial Fulfilment of the Requirements of the Master Degree in Computer Science. <https://meu.edu.jo/libraryTheses/How%20to%20Detect%20Phishing%20Website.pdf>

*[12] Towards benchmark datasets for machine learning based website phishing detection: An experimental study - Lay Summaries - Engineering Applications of Artificial Intelligence - Journal - Elsevier*. (2021). Towards Benchmark Datasets for Machine Learning Based Website Phishing Detection: An Experimental Study. <https://www.journals.elsevier.com/engineering-applications-of-artificial-intelligence/lay-summaries/towards-benchmark-datasets-for-machine-learning-based-website-phishing-detection-an-experimental-study>

*[13] Using Lexical Features for Malicious URL Detection- A Machine Learning Approach*. (2019, November 14). [Video]. YouTube. <https://www.youtube.com/watch?v=MIR2RJZjAOg>