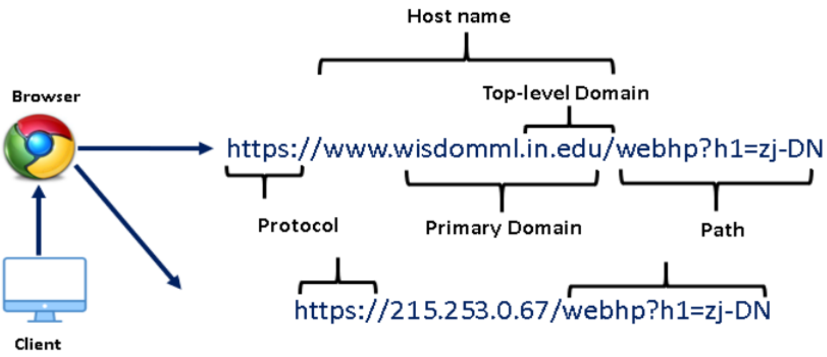
Malicious URL detection using Machine Learning and Artificial Intelligence


Host name
Browser
Top-level Domain
https://www.wisdomml.in.edu/webhp?h1=zj-DN
Protocol
Primary Domain
Path
https://215.253.0.67/webhp?h1=zj-DN
Client

Author: Pranalee Peshne

June 12th, 2023

Berkeley Engineering | BerkeleyHaas

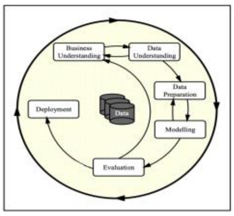
Overview:

The Web has long become a major platform for online criminal activities. URLs are used as the main vehicle in this domain. To counter this issues security community focused its efforts on developing techniques for mostly blacklisting of malicious URLs. While successful in protecting users from known malicious domains, this approach only solves part of the problem. The new malicious URLs that sprang up all over the web in masses commonly get a head start in this race. .

In this project I explored a lightweight approach to detection and categorization of the malicious URLs according to their attack type and show that lexical analysis is effective and efficient for proactive detection of these URLs.

Methodology:

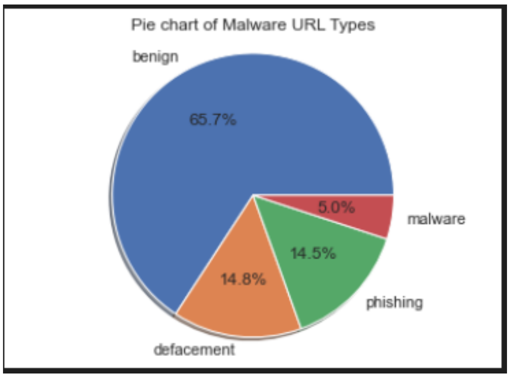
The Cross Industry Standard Process for Data Mining (CRISP-DM) is a process model that serves as the base for a data science process. It has six sequential phases: The project was organized as per the below the CRISP-DM Framework phases.



- Business understanding – What does the business need?
- Data understanding – What data do we have / need? Is it clean?
- Data preparation – How do we organize the data for modelling?
- Modelling – What modelling techniques should we apply?
- Evaluation – Which model best meets the business objectives?
- Deployment – How do stakeholders access the results?

Dataset:

This project is using a Malicious URLs dataset of 6,51,191 URLs, out of which 4,28,103 benign or safe URLs, 96,457 defacement URLs, 94,111 phishing URLs, and 32,520 malware URLs. The different types of URLs in the dataset are: Benign, Malware, Phishing, and Defacement URLs.



- Benign URLs: These are safe to browse URLs. Some of the examples of benign URLs are as follows:

  mp3raid.com/music/krizz_kaliko.htmlinfinitysw.comgoogle.co.inmyspace.com

- Malware URLs: These type of URLs inject malware into the victim's system once he/she visit such URLs. Some of the examples of malware URLs are as follows:

  proplast.co.nz

  http://103.112.226.142:36308/Mozi.mmicroencapsulation.readmyweather.com

xo3fhvm5lcvzy92q.download

- Defacement URLs: Defacement URLs are generally created by hackers with the intention of breaking into a web server and replacing the hosted website with one of their own, using techniques such as code injection, cross-site scripting, etc. Common targets of defacement URLs are religious websites, government websites, bank websites, and corporate websites. Some of the examples of defacement URLs are as follows:

  http://www.vnic.co/khach-hang.html

http://www.raci.it/component/user/reset.html

- Phishing URLs: By creating phishing URLs, hackers try to steal sensitive personal or financial information such as login credentials, credit card numbers, internet banking details, etc. Some of the examples of phishing URLs are shown below:

  roverslands.net

  -- corporacionrossenditotours.com

  -- http://drive-google-com.fanalav.com/6a7ec96d6a

Dataset Features:

The Malicious URL dataset was taken from [Kaggle](https://www.kaggle.com/datasets/sid321axn/malicious-urls-dataset)

The dataset includes a large number of examples of Malicious URLs so that a machine learning-based model can be developed to identify malicious URLs and stop them in advance before infecting computer system or spreading through intentioned. The Kaggle dataset is pre-processed dataset, the original source of data is from [Canadian Institute for Cybersecurity] (https://www.unb.ca/cic/datasets/url-2016.html). For increasing phishing and malware URLs, a Malware domain blacklist dataset was used. To increased benign URLs Faizan git repo was used. Phishing URLs were increased using PhishTank dataset and PhishStorm dataset. In nutshell the dataset used in this project is collected from different sources. The URLs were collected from different sources into a separate data frame and finally merge them to retain only URLs and their class type.

The dataset features contained in the maliciousurl_input.csv:

- url = Malacious URL

- type =Type of URL benign, malware,defacement,phishing

- category= LabelEncoded URL type

The data file maliciousurl_input.csv contain two column “url” and “type”.

Using data pre-processing techniques following features were extracted.

- url_len = Length of URL
- domain =Extracted domain name from the URL
- root_domain = root domain of the URL
- count_at =count of at special char
- count-question =count of ? special char
- count_hyphen =count of - special char
- count_equal =count of = special char
- count_dot =count of . special char
- count_hash=count of # special char
- count_percent =count of % special char
- count_plus=count of + special char
- count_dollarsign=count of $ special char
- count_exlamation =count of ! special char
- count_star =count of * special char
- count_comma=count of , special char
- count_double_slash=count of // special char
- count_slash=count of single / special char
- abnormal_url =count of abnormal urls
- count-https= count of https
- count-http = count of http
- count-www = count of www
- digits_count count of digits
- hostname_length =count of hostname
- sus_url =count of suspicious urls

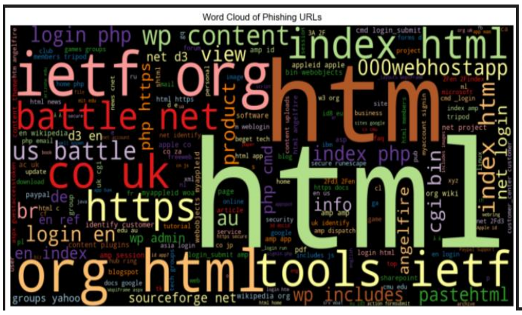
- letters_count= letter count

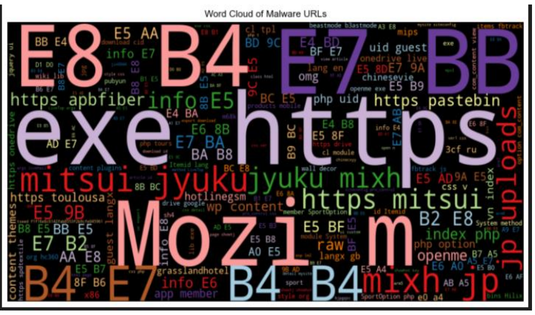
- short_url= short url count

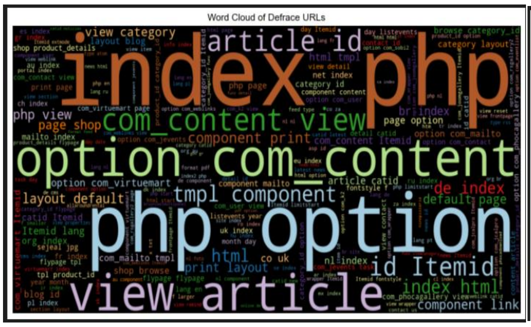
- use_of_ip = finp if ip address is present in the URL

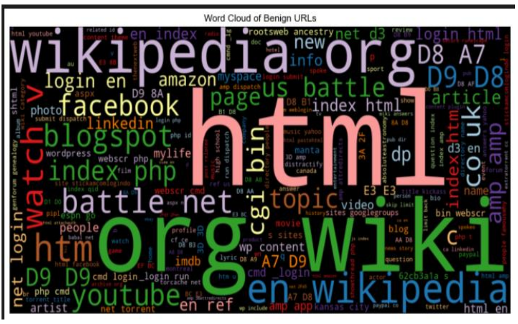
Exploratory Data Analysis (EDA):

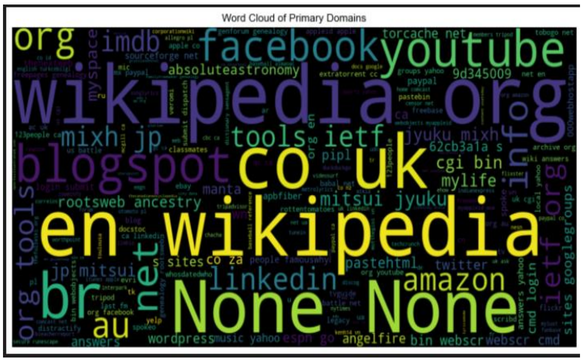
Word cloud of URLs: The word cloud was used to view the pattern of words/tokens in particular target label. From the below word clouds images, most of the tokens distribution in different types of URLs is now crystal clear.

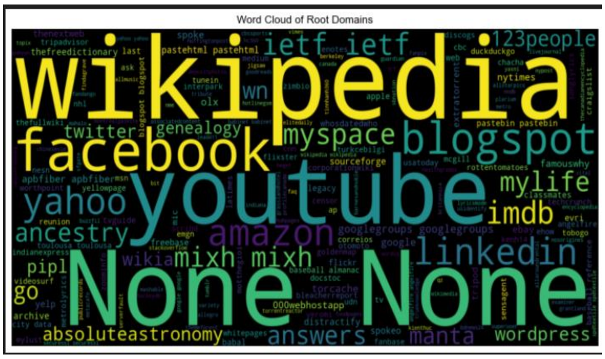

Word Cloud of Phishing URLs

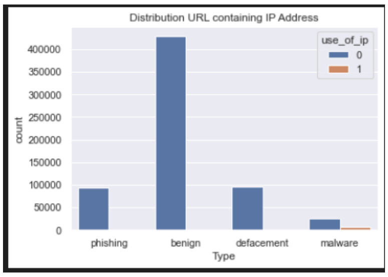

Word Cloud of Malware URLs

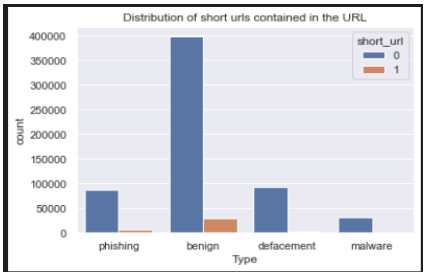

Word Cloud of Defrace URLs

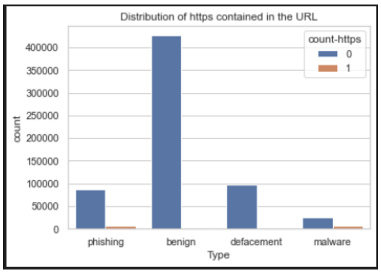

Word Cloud of Benign URLs

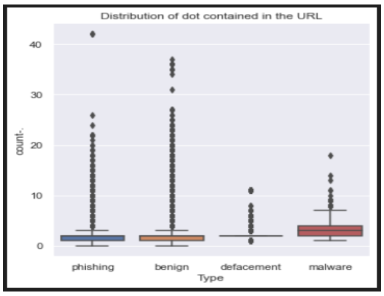

Word Cloud of Primary Domains

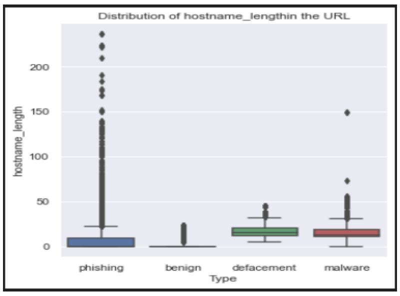

Word Cloud of Root Domains





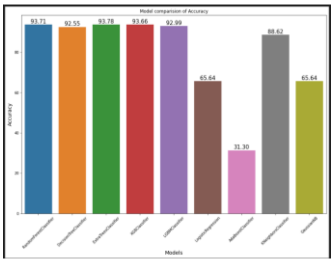


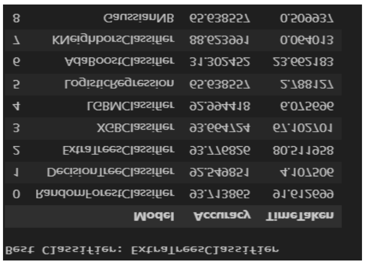




Model Exploration:

- Baseline Model: Logistic Regression: Poor accuracy but reasonable time taken to fit.
- Multi model exploration: Following model were explored to identify model with the best score for the pre-processed Malware URL dataset. RandomForestClassifier, DecisionTreeClassifier, ExtraTreesClassifier,XGBClassifier,LGBMClassifier,LogisticRegression,AdaBoostClassifier,KNeighborsClassifier,GaussianNB. Other traditional models were exploring but dropped as they were taking too much time with low accuracy.



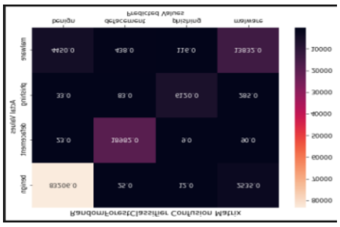


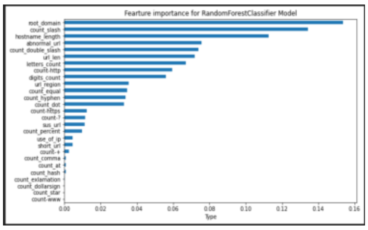
```
Best Classifier: ExtraTreesClassifier

                    Model   Accuracy  TimeTaken
0  RandomForestClassifier  93.713865  91.612999
1  DecisionTreeClassifier  92.549851   4.107506
2    ExtraTreesClassifier  93.776826  80.511958
3           XGBClassifier  93.664724  67.102701
4          LGBMClassifier  92.994418   6.075696
5      LogisticRegression  65.638557   2.788127
6      AdaBoostClassifier  31.302452  23.662183
7    KNeighborsClassifier  88.623991   0.064013
8              GaussianNB  65.638557   0.509937
```

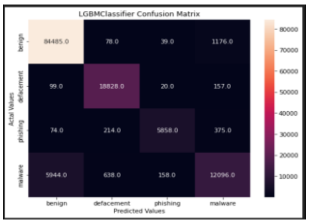
- RandomForestClassifier, LGBMClassifier, ExtraTreesClassifier, XGBClassifier model were further selected. Feature importance was determined. RandomForest and Light GBM Classifier gave the best accuracy, f1 score, recall and use of features. Hence

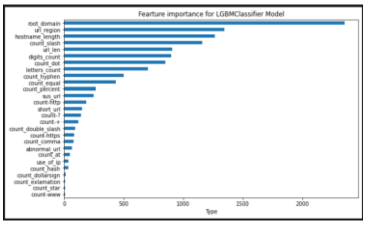
RandomForestClassifier and LGBMClassifier were further selected for hyperparameter evaluation.

- RandomForestClassifier and LGBMClassifier were the selected model, hyperparameter were further explored using GrisSerachCV and RandomSearchCV









- Neural Net Karas TensorFlow Sequentian and KerasClassifier with GridSearchCV were explored. As the Keras model gave low accuracy they were not considered. Keras sequential model gave be accuracy of 41.32% and loss -3747731017302016.00

Next Steps:

- Experiment further with hyperparameter to identity the best fit for the selected model
- Put the model to practical use.
- Explore deploying the model to MLOps Platform such as AWS Sage Maker, Azure ML and Google Cloud ML

References:

Course: UC BERKELEY Engineering and Haas Professional Certificate in Machine Learning & Artificial Intelligence course content, tutorial, videos, etc.
Home - Karas Documentation - https://keras.io/
TensorFlow - https://www.tensorflow.org/
SKlearn|https://scikit-learn.org/
Kaggle|https://www.kaggle.com/code/thisishusseinali/malicious-url-detection
Canadian Institute for Cyber |https://www.unb.ca/cic/datasets/url-2016.htm

Online Examples|https://github.com/Colorado-Mesa-University-Cybersecurity

People: Jessica Cervi, Savio Saldanha, Holly Bees, and Leanna Biddle, and course students