Malicious URL Detection

— Using Machine Learning

Data Mining & Predictive Analytics - BUDT758T

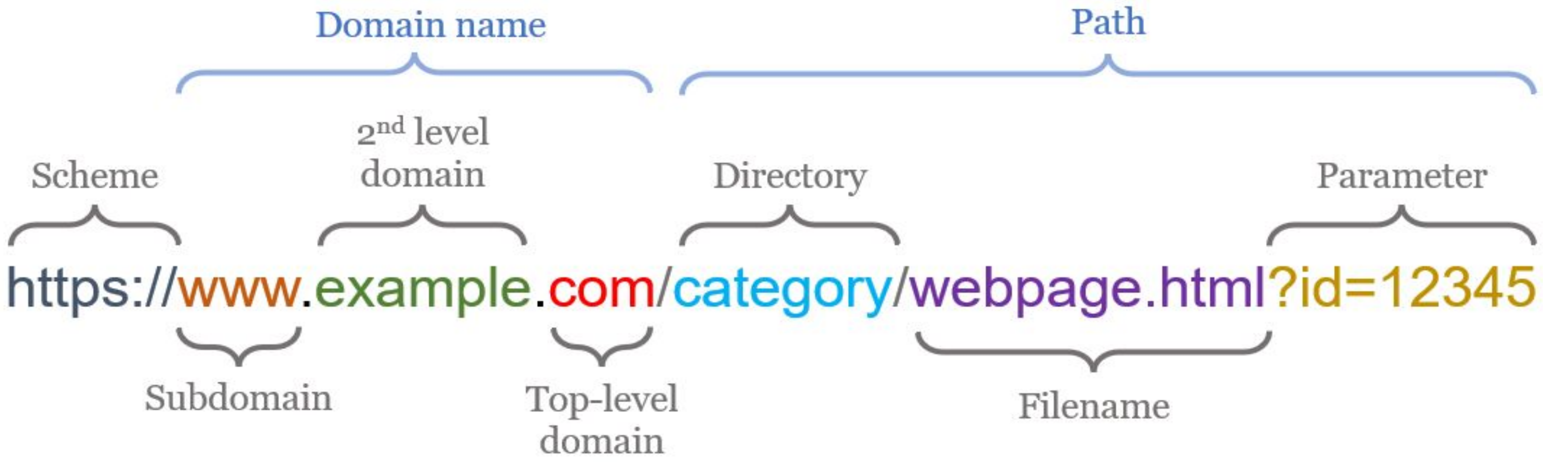
Group: Alekya Ghanta, Weian Shi, Xingjian Qin, Yanpu Wang, Jie Gao

Agenda

1. Introduction & Research Questions
2. Data Collection
3. Preprocessing
4. EDA
5. Models
6. Conclusion

What is a URL?

Uniform Resource Locator is a web address that identifies the location of a web page, a file, or a resource on the World Wide Web.

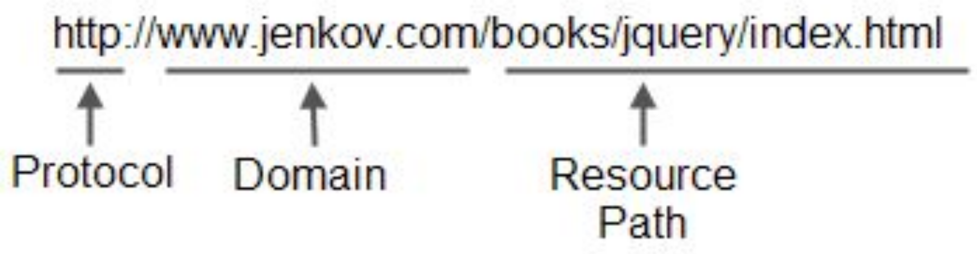


What is a URL?

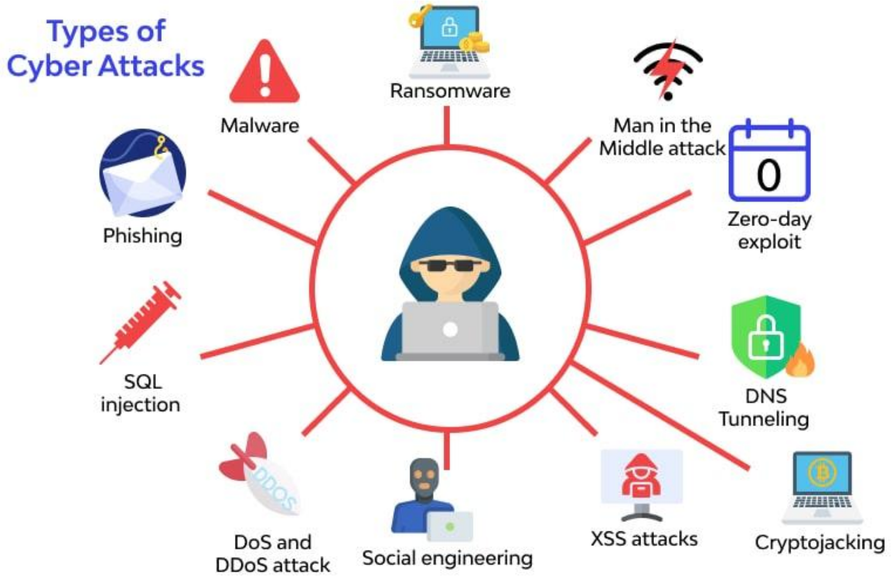
Scheme/Protocol: Specifies the method used to access the resource. The most common protocols used in URLs are HTTP (Hypertext Transfer Protocol) and HTTPS (HTTP Secure)

Domain: Identifies one or more IP addresses on the internet. Domain is the name used to identify a website. Rather than remembering the IP, users can simply type the domain name to access the website.

Path: Part of the URL that follows the domain name and identifies the specific file that the resource is located in. The path can include multiple directories, subdirectories, and file names, separated by ("/").



Threats associated with URLs



There are several threats associated with URLs that users should be aware of. However, we are limiting the scope of this project to **Malware and Phishing** attacks.

Phishing: Attackers can **create fake URLs** that appear to be legitimate ones to trick users into giving sensitive information such as **login credentials** or **bank details**.

Malware: URLs can be used to deliver viruses, trojans to a user's device for **stealing personal information, damaging computer's hardware**. Malware can be in the URL itself or on the website that the URL leads to.

Research Area of Focus

Develop an accurate and efficient **machine learning model** that can classify URLs as safe, malware or phishing with high accuracy.

Identify **potentially dangerous URLs** and prevent users from accessing them, thereby reducing the **risk of cybersecurity** threats.

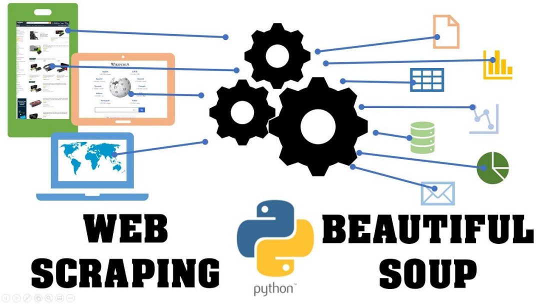
Compare the **performance** of various **algorithms**, as well as the **impact of different features** on the classification accuracy.

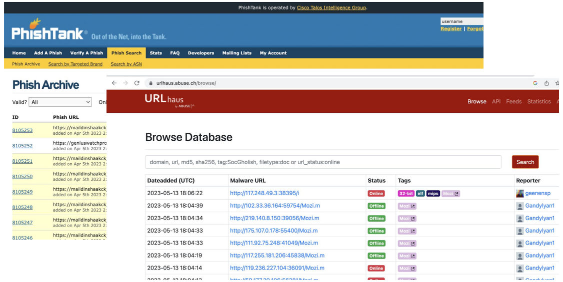
Data Collection

Sources

Collected **Phish & safe URLs** from PhishTank, a community-based phish verification system where users submit suspected phishes and other users "vote" if it is a phish or not)

Collected **Malware URLs** from URLhaus a project operated by abuse.ch to collect, track, and share malware URLs, helping network administrators & security analysts to protect their networks from cyber threats.





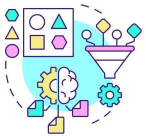
Challenges

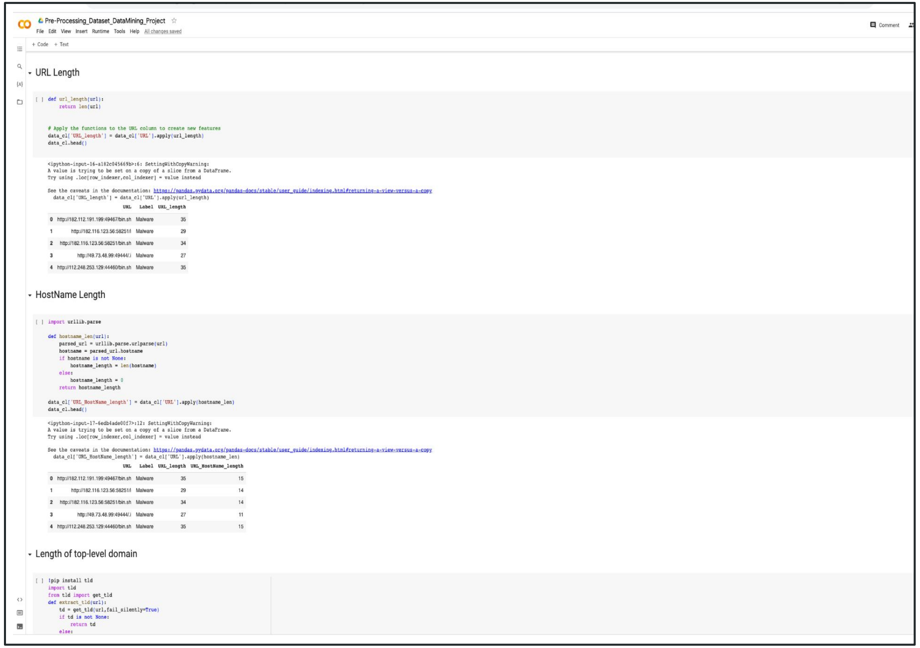
- **Limited data import speed** – multiple requests may lead to server blocking or flagging the scraper as suspicious activity.
- **Data Size** - Over 1GB Data, too much to handle.
- **Class Imbalance** – Number of Safe URLs in the extracted dataset were just 25 for the time period.

Solutions

- **Delay** - Used time lapse to avoid too many requests at same time.
- Limited the time frame of data import from **April 1st – May 1st**
- Extracted sufficient Safe URLs instead of making genetic copies to better train the model.
- Ran over multiple systems & combined the data in **csv format.**

Data Preprocessing - Feature Engineering





Pre-Processing_Dataset_DataMining_Project

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

Comment

URL Length

```
def url_length(url):
    return len(url)


# Apply the functions to the URL column to create new features
data_cl['URL_length'] = data_cl['URL'].apply(url_length)
data_cl.head()
```

```
<ipython-input-16-a182c045669b>:6: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  data_cl['URL_length'] = data_cl['URL'].apply(url_length)
```

	URL	Label	URL_length
0	http://182.112.191.199:49467/bin.sh	Malware	35
1	http://182.116.123.56:58251/i	Malware	29
2	http://182.116.123.56:58251/bin.sh	Malware	34
3	http://49.73.48.99:49444/.i	Malware	27
4	http://112.248.253.129:44460/bin.sh	Malware	35

HostName Length

```
import urllib.parse

def hostname_len(url):
    parsed_url = urllib.parse.urlparse(url)
    hostname = parsed_url.hostname
    if hostname is not None:
        hostname_length = len(hostname)
    else:
        hostname_length = 0
    return hostname_length

data_cl['URL_HostName_length'] = data_cl['URL'].apply(hostname_len)
data_cl.head()
```

```
<ipython-input-17-6edb4ade00f7>:12: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  data_cl['URL_HostName_length'] = data_cl['URL'].apply(hostname_len)
```

	URL	Label	URL_length	URL_HostName_length
0	http://182.112.191.199:49467/bin.sh	Malware	35	15
1	http://182.116.123.56:58251/i	Malware	29	14
2	http://182.116.123.56:58251/bin.sh	Malware	34	14
3	http://49.73.48.99:49444/.i	Malware	27	11
4	http://112.248.253.129:44460/bin.sh	Malware	35	15

Length of top-level domain

```
!pip install tld
import tld
from tld import get_tld
def extract_tld(url):
    td = get_tld(url,fail_silently=True)
    if td is not None:
        return td
    else:
```

15 Features have been extracted from the raw URLs to develop a predictive model which can classify URLs into Phish, Malware & Safe URLs

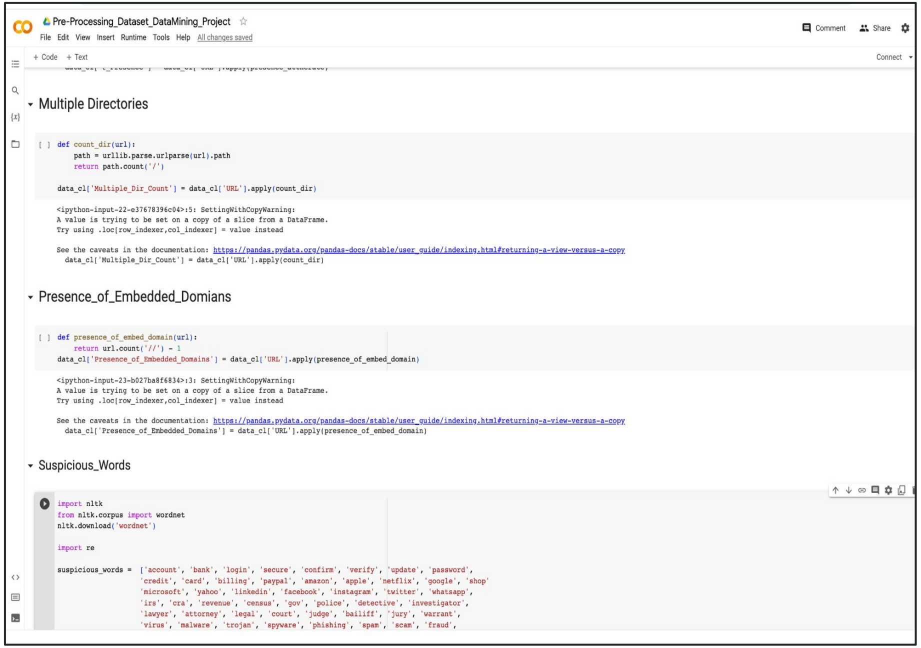
Features based on Length :

- URL Length
- HostName Length
- Length of Top Level Domain

Features based on Abnormality : Presence of

- Multiple Domains
- Multiple www's
- Multiple Directories

Data Preprocessing - Feature Engineering



Pre-Processing_Dataset_DataMining_Project

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

Comment Share

Connect

Multiple Directories

```
def count_dir(url):
    path = urllib.parse.urlparse(url).path
    return path.count('/')

data_cl['Multiple_Dir_Count'] = data_cl['URL'].apply(count_dir)
```

```
<ipython-input-22-e37678396c04>:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  data_cl['Multiple_Dir_Count'] = data_cl['URL'].apply(count_dir)
```

Presence_of_Embedded_Domians

```
def presence_of_embed_domain(url):
    return url.count('//') - 1
data_cl['Presence_of_Embedded_Domains'] = data_cl['URL'].apply(presence_of_embed_domain)
```

```
<ipython-input-23-b027ba8f6834>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  data_cl['Presence_of_Embedded_Domains'] = data_cl['URL'].apply(presence_of_embed_domain)
```

Suspicious_Words

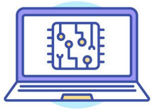
```
import nltk
from nltk.corpus import wordnet
nltk.download('wordnet')

import re

suspicious_words =  ['account', 'bank', 'login', 'secure', 'confirm', 'verify', 'update', 'password',
                     'credit', 'card', 'billing', 'paypal', 'amazon', 'apple', 'netflix', 'google', 'shop'
                     'microsoft', 'yahoo', 'linkedin', 'facebook', 'instagram', 'twitter', 'whatsapp',
                     'irs', 'cra', 'revenue', 'census', 'gov', 'police', 'detective', 'investigator',
                     'lawyer', 'attorney', 'legal', 'court', 'judge', 'bailiff', 'jury', 'warrant',
                     'virus', 'malware', 'trojan', 'spyware', 'phishing', 'spam', 'scam', 'fraud',
```

Features based on Abnormality : (contd)

- Embedded domains
- Multiple HTTP & Presence of HTTPS
- Use of URL Shortening
- Digit Count
- Suspicious words

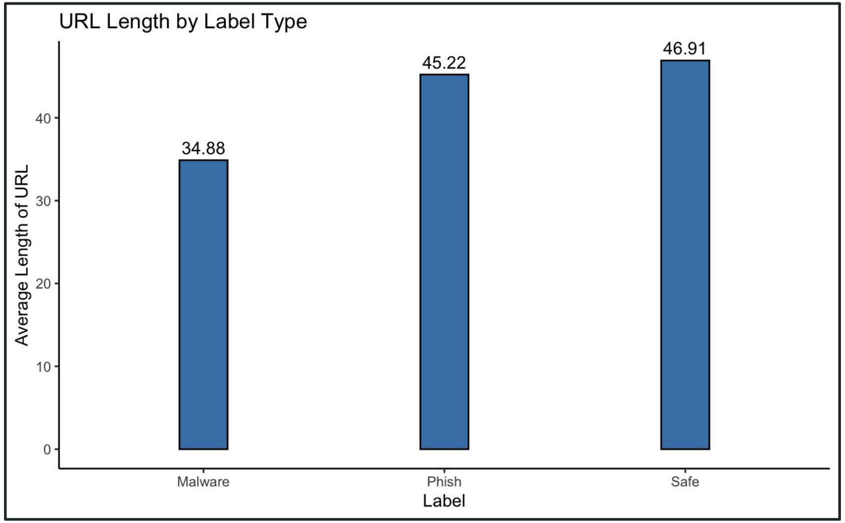


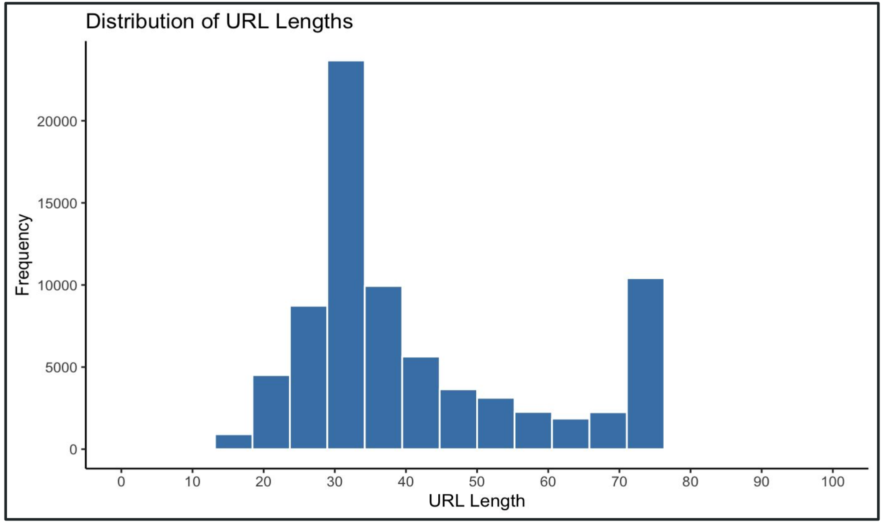
Features based on Special Characters:

- Count of ?
- Count of =
- Presence of @

Exploratory Data Analysis

Average length of Safe URLs is highest unlike the usual expectations. Most of the URLs fall in the 30-35 length bin.

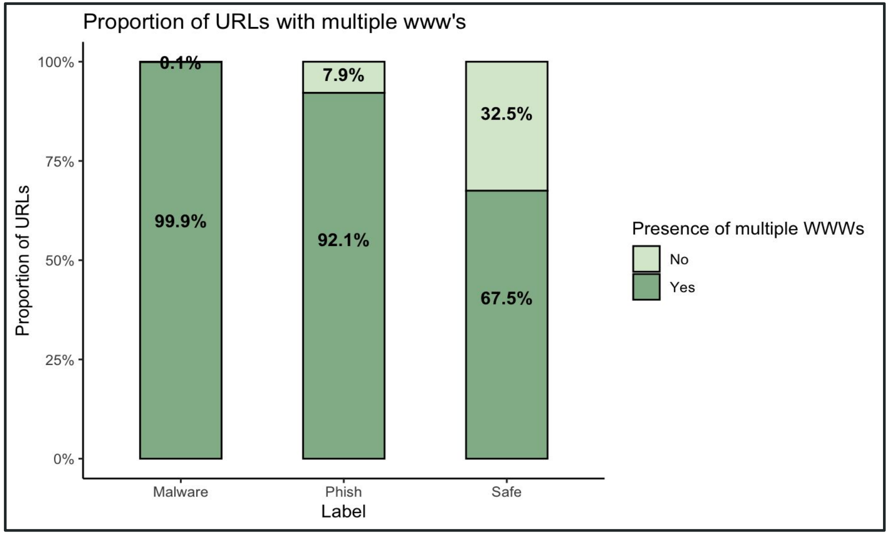


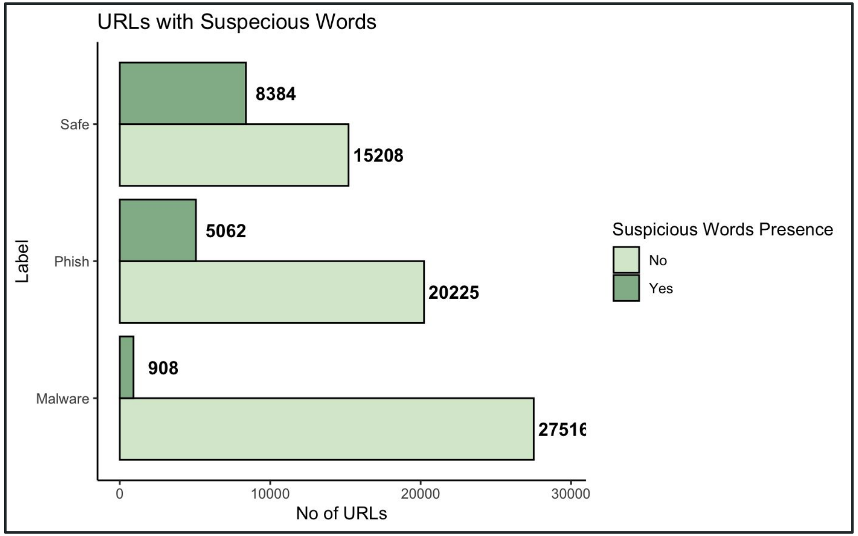


Introduction | Data Collection | Preprocessing | EDA | Models | Conclusion

Exploratory Data Analysis

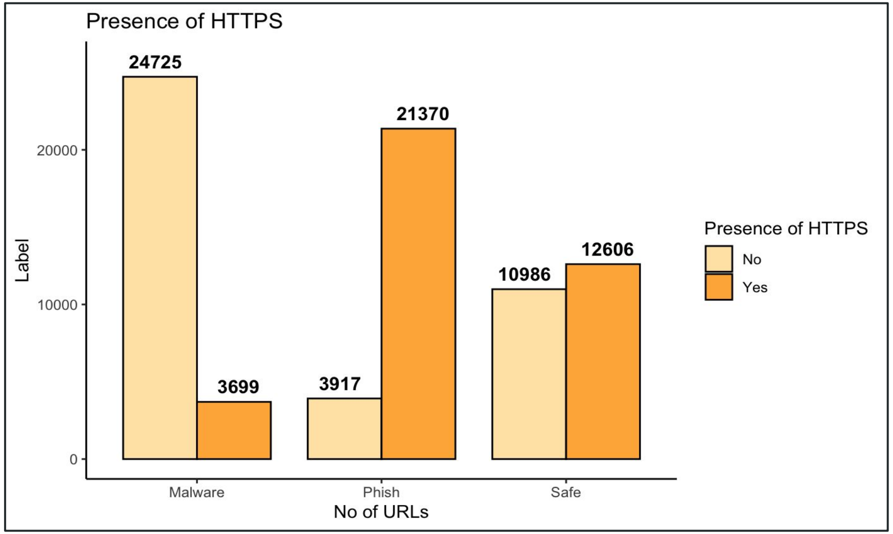
Malware URLs mostly have multiple WWWs. Phish URLs tend to have dubious words when compared to Malware URLs

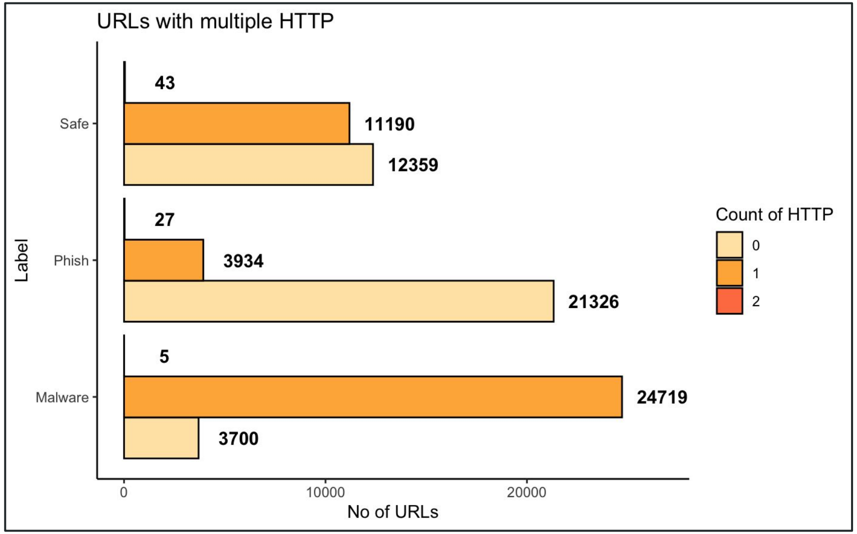




Exploratory Data Analysis

Most of the Malware URLs have no Secure Protocol & presence of Multiple HTTPs in the URL text.





Predictive Model: Naive Bayes

Metric	*Malware*	*Phish*	*Safe*
Sensitivity	0.85	0.84	0.44
Specificity	0.89	0.75	0.94
Positive Pred Value	0.82	0.63	0.75
Negative Pred Value	0.91	0.90	0.79
Prevalence	0.37	0.33	0.30
Detection Rate	0.31	0.28	0.13
Detection Prevalence	0.38	0.44	0.17
Balanced Accuracy	0.87	0.80	0.69

Accuracy : 72.2%

Predictive Model: Decision Tree

Metric	*Malware*	*Phish*	*Safe*
Sensitivity	0.98	0.70	0.51
Specificity	0.90	0.82	0.90
Positive Pred Value	0.86	0.65	0.68
Negative Pred Value	0.99	0.84	0.81
Prevalence	0.37	0.33	0.30
Detection Rate	0.36	0.23	0.15
Detection Prevalence	0.42	0.35	0.23
Balanced Accuracy	0.94	0.76	0.71

Accuracy : 74.6%

Predictive Model: Gradient Boosting Model

Metric	*Malware*	*Phish*	*Safe*
Sensitivity	0.90	0.78	0.76
Specificity	0.97	0.87	0.89
Positive Pred Value	0.95	0.74	0.75
Negative Pred Value	0.94	0.89	0.90
Prevalence	0.37	0.33	0.30
Detection Rate	0.33	0.26	0.23
Detection Prevalence	0.35	0.34	0.31
Balanced Accuracy	0.93	0.82	0.82

Accuracy : 81.7%

Predictive Model: Random Forest - Bagging

Metric	*Malware*	*Phish*	*Safe*
Sensitivity	0.97	0.85	0.84
Specificity	0.98	0.93	0.93
Positive Pred Value	0.97	0.86	0.83
Negative Pred Value	0.98	0.93	0.93
Prevalence	0.37	0.33	0.30
Detection Rate	0.36	0.28	0.25
Detection Prevalence	0.37	0.33	0.30
Balanced Accuracy	0.98	0.89	0.88

Accuracy : 89.3%

Predictive Model: Random Forest

Metric (m=8)	*Malware*	*Phish*	*Safe*
Sensitivity	0.98	0.85	0.84
Specificity	0.98	0.94	0.93
Positive Pred Value	0.96	0.87	0.84
Negative Pred Value	0.99	0.93	0.93
Prevalence	0.37	0.33	0.30
Detection Rate	0.36	0.28	0.25
Detection Prevalence	0.37	0.32	0.30
Balanced Accuracy	0.98	0.89	0.89

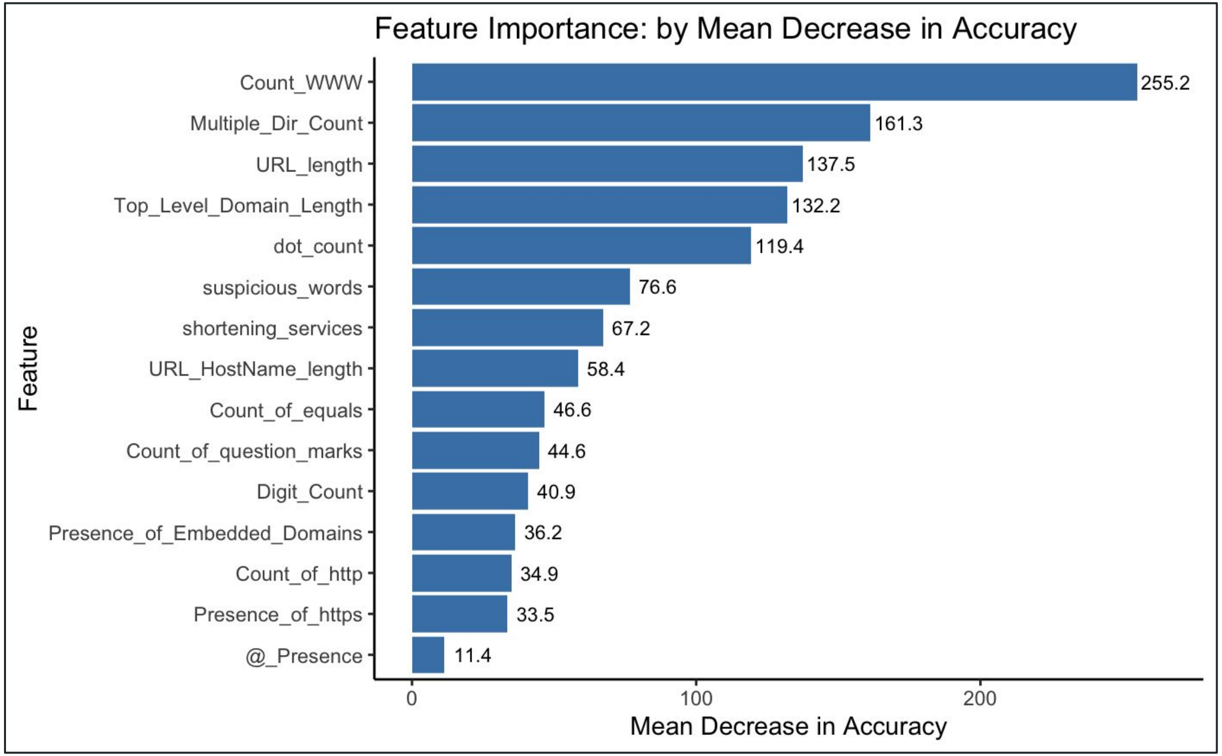
Accuracy : 89.7%

Model Performance Comparison

Model	*Accuracy*	*FPR (Classifying Safe URL as Unsafe)*	*FNR (Classifying Unsafe URL as Safe)*
Naive Bayes	**72.2%**	**56.3%**	**6.3%**
Decision Tree	**74.6%**	**48.4%**	**10.3%**
Gradient Boosting	**81.7%**	**23.8%**	**11.1%**
Random Forest - Bagging	**89.3%**	**16.7%**	**7.1%**
Random Forest (m=8)	**89.7%**	**15.8%**	**6.7%**

Random Forest performs well Overall, in terms of **Accuracy, FPR & FNR**. However, **Naive Bayes** model also turns out to be an important model as False Negatives are much costlier than False Positives in this case.

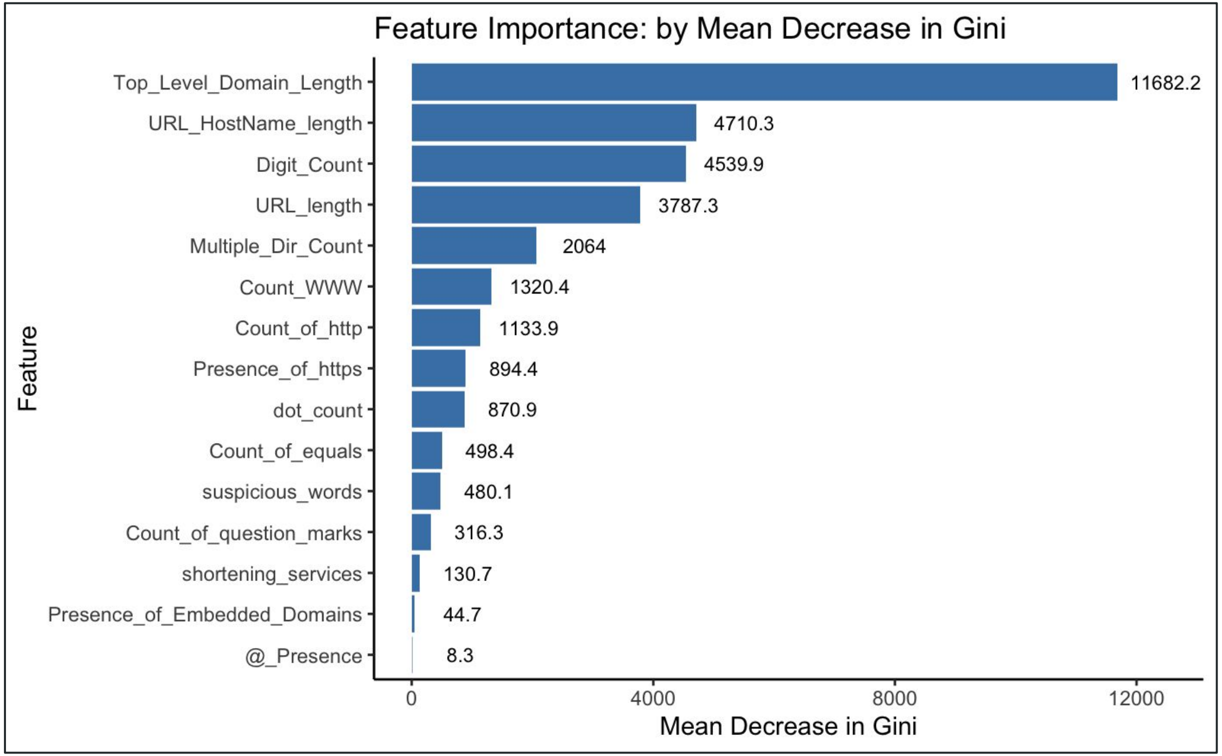
Feature Importance



A **high MDA** value suggests that the feature is more **important** in predicting the target variable.

Count of WWWs, Multiple Directories count, URL Length, Top Domain Length, multiple domain count seem to be the **top 5 features** which help in classifying the URLs into safe, phish & malware.

Feature Importance



The **mean decrease in Gini importance** of a feature is calculated by summing up total **decrease in Gini index** for each node that **splits on the feature**, then taking the average across all trees in the random forest.

Higher decrease in mean Gini index for a feature indicates that the feature is **more important** in splitting the data and creating a decision tree that accurately predicts the target variable.

Top Level Domain Length, Digit Count, Host Name Length, URL Length, Multiple Directory count are the **top 5** important features.

Conclusion

- Average length of Safe URLs is highest unlike the usual expectations. Most of the URLs fall in the 30-35 length bin.
- Malware URLs mostly have **multiple WWWs**. Phish URLs tend to have **dubious words** when compared to Malware URLs.
- Most of the Malware URLs have **no Secure Protocol** & presence of **Multiple HTTPs** in the URL text.
- **Random Forest** performs well Overall, in terms of **Accuracy, FPR & FNR**
- **Naive Bayes** model also turns out to be an important model as False Negatives are more costlier than False Positives in this case.
- Count of WWWs, Multiple Directories count, URL Length, Top Domain Length, multiple domain count seem to be the **top 5 features** by **Mean Decrease in Accuracy.**
- Top Level Domain Length, Digit Count, Host Name Length, URL Length, Multiple Directory count are the **top 5** important features by **Mean Decrease in Gini.**

References

- https://wisdomml.in/malicious-url-detection-using-machine-learning-in-python/
- https://www.kaggle.com/datasets/shashwatwork/web-page-phishing-detection-dataset
- https://siddharthakancharla.medium.com/classification-of-urls-f8253dee914
- https://www.kaggle.com/datasets/sid321axn/malicious-urls-dataset