Malicious URL Detection, classifier and web security Analyzer

B. TECH SEM – VII Minor PROJECT
Dept. of Computer Science & Engineering

By

Sahil Panchasara 19BCP115
Shubham Kathiriya 19BCP127

Under the Supervision
Of
Dr. Chintan Bhatt
Dr. Kaushal Shah

SCHOOL OF TECHNOLOGY
PANDIT DEENDAYAL ENERGY UNIVERSITY
GANDHINAGAR, GUJARAT, INDIA
July – December, 2022

Abstract

Cybersecurity is seriously threatened by scam websites or URLs. Malicious URLs serve unsolicited content (spam, phishing, drive-by downloads, etc.) and deceive users into falling for schemes, resulting in billions of dollars in losses every year (including financial loss, identity theft, and malware installation). In order to create a machine learning-based model to recognize dangerous URLs and prevent them from destroying computer systems or propagating across the internet, we have assembled this dataset to include a substantial number of samples of malicious URLs. We have developed a machine learning model for detection and classification. For detection we used Logistic Regression with the testing accuracy of nearly 99.6% and for classification we used Decision Tree with the testing accuracy of nearly 91%. We have also made a web extension as a result of practical implementation of our work.

Table of Contents

Introduction

Phishing has recently risen to the top of security professionals' concerns due to how simple it is to create a phony website that looks remarkably similar to an actual one. Although professionals can recognize fraudulent websites, not all users can, which is why some customers become victims of phishing schemes. The main objective of the attacker is to get bank account passwords. Customers falling for phishing schemes cost American businesses $2 billion per year. The third Microsoft Computing Safer Index Report, which was released in February 2014, estimates that phishing may cost more than $5 billion annually throughout the world. Users are increasingly becoming targets because they are unaware of phishing attacks. Since phishing attacks feed on user vulnerabilities, it is extremely difficult to stop them, yet it is essential to develop phishing detection systems. Attackers use inventive tactics to trick people, such as obfuscation, fast-flux, where proxies are created instantly to host the website, algorithmic synthesis of new URLs, etc., to alter the URL and make it seem real. In order to recognize phishing websites, the "blacklist" method involves adding prohibited URLs and Internet Protocol (IP) addresses to the antivirus database.

Heuristic-based detection, which considers traits that have been confirmed to occur in actual phishing assaults, is capable of detecting zero-hour phishing efforts even though the attributes are not always guaranteed to be present in such attacks and the false positive rate for detection is relatively high. Many security professionals are focusing more on machine learning techniques to get beyond the limitations of blacklist and heuristics-based tactics. Many machine learning algorithms base their conclusions or decisions on previous information. An algorithm will assess the features of a massive number of valid and prohibited URLs in order to successfully identify phishing websites, including zero-hour phishing websites. Computer users can quickly and securely enter this type of false website, which is thought to be the real website, while entering their crucial information. due to the fact that it appears that the web page entered is an exact duplicate of the original web page. According to a related study on how individuals react to phishing attempts, there are five primary reasons why computer users fall for phishing:

- Users lack a thorough understanding of URLs.
- Users are unaware of which websites are reliable.
- Due to URL masking or redirection, users cannot see the full address of the web page.
- Users frequently lack the time to carefully read the URL, which can result in them accidentally accessing some websites.
- Users lack the ability to differentiate between legitimate and fake websites.

Academics have recently paid a lot of attention to phishing attempts, which are sometimes likened to "fishing" for victims. Attackers (also known as phishers) find it to be a successful and alluring strategy to create specialized fake websites with exact designs that resemble well-known and reliable websites on the Internet. Despite having identical graphical user interfaces, each of these webpages must have a different URL from the original website. By examining their URLs, a cautious and knowledgeable user can easily identify these rogue web pages. Due to the rapid pace of life, end consumers seldom examine the complete address of an active online page, which is frequently provided by other websites, social networking sites, or even just a simple email message. By using these kinds of phony URLs, a phisher tries to get sensitive and private information about the victim, including financial information, personal information, usernames, passwords, etc.

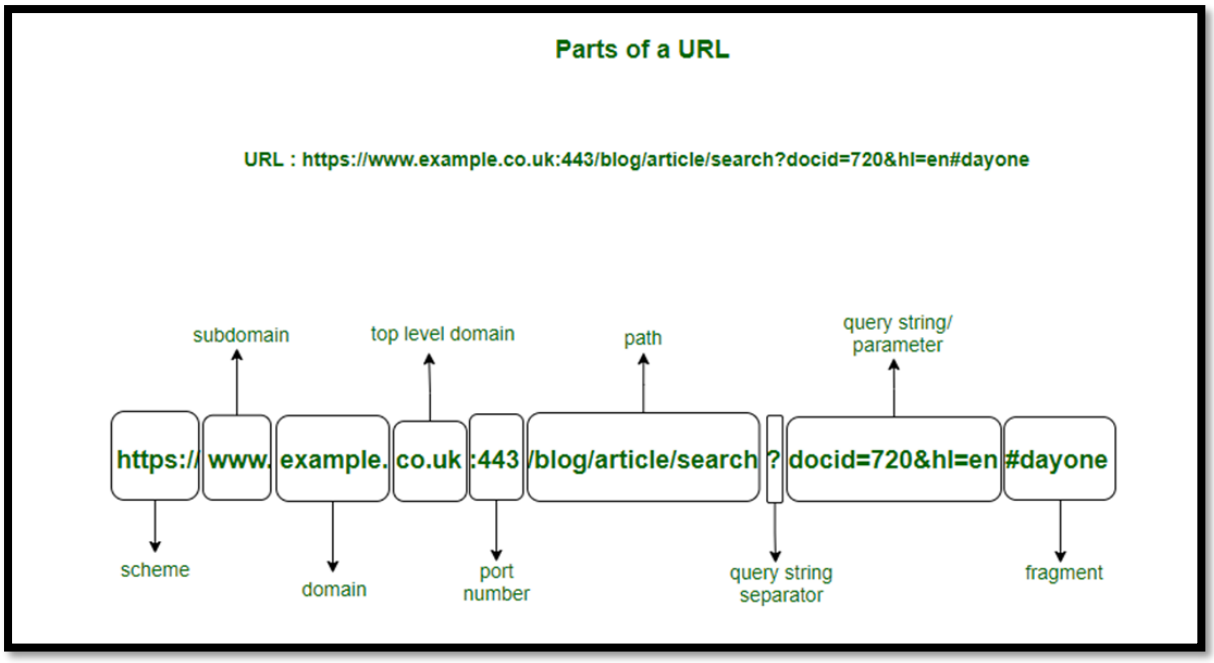


Figure 1: Parts of a URL

For locating and recording vulnerabilities, there is a standard called Common Vulnerability Exposure (CVE). Researchers can trace security issues over time and locate them all around the world thanks to this approach. As a result, it will be easier for users, software developers, and other parties to keep track of the appearance of specific security issues in code. Several risk categories, such as operational risk and control risk, have already been discussed. The CVE focuses on the potential for software to be vulnerable to cyberattacks, making it a distinct form of risk. This article defines a CVE, explains how it works, and explores why it's crucial for your company's strategy to manage cyber risk.

To keep track of flaws in software and other items, a CVE, or Common Vulnerabilities and Exposures, is employed. Researchers can maintain track of previously identified vulnerabilities using the CVE and prevent them from being reported as newly discovered vulnerabilities. The CVE is administered by the MITRE Corporation, a nonprofit organization that oversees several standards and monitoring systems for the public and corporate sectors. It's critical to comprehend the following three CVE components: They have backwards compatibility, are unique, and are standardized. To put it another way, every CVE is unique, every CVE can be used to patch existing vulnerabilities, and every CVE follows a predetermined pattern to ensure consistency. This is useful for tracking vulnerabilities over time and for comparing the severity of different vulnerabilities.

Literature Review

Because of the rapid expansion of the Internet, consumers are moving their shopping habits away from traditional retail and toward online shopping. Nowadays, burglars search for victims online using specific methods rather than stealing stores like supermarkets and banks. Hackers have created cutting-edge techniques like phishing to deceive users into entering personal information, such as account IDs, usernames, and passwords, on phony websites because of the anonymity of the Internet. Determining if a web page is genuine or fake is a very challenging process since phishing uses a semantics-based attack strategy that primarily preys on computer users' vulnerabilities.
Software companies are introducing blacklists, heuristics, visual, and machine learning-based anti-phishing solutions, but they can't thwart every assault. The real-time anti-phishing system presented in their work uses seven different categorization algorithms in addition to features based on natural language processing (NLP). The system differs from other studies in the literature in the following ways: it does not depend on any third-party services, executes in real-time, uses a large amount of real and phishing data, can identify new websites, and uses feature-rich classifiers. The performance of the system is measured using a brand-new dataset, which is also utilized to assess how well the experiment worked. According to testing and comparison data, the Random Forest technique surpasses the other implemented classification algorithms, with an accuracy rate of about 98% for phishing URL recognition [1].

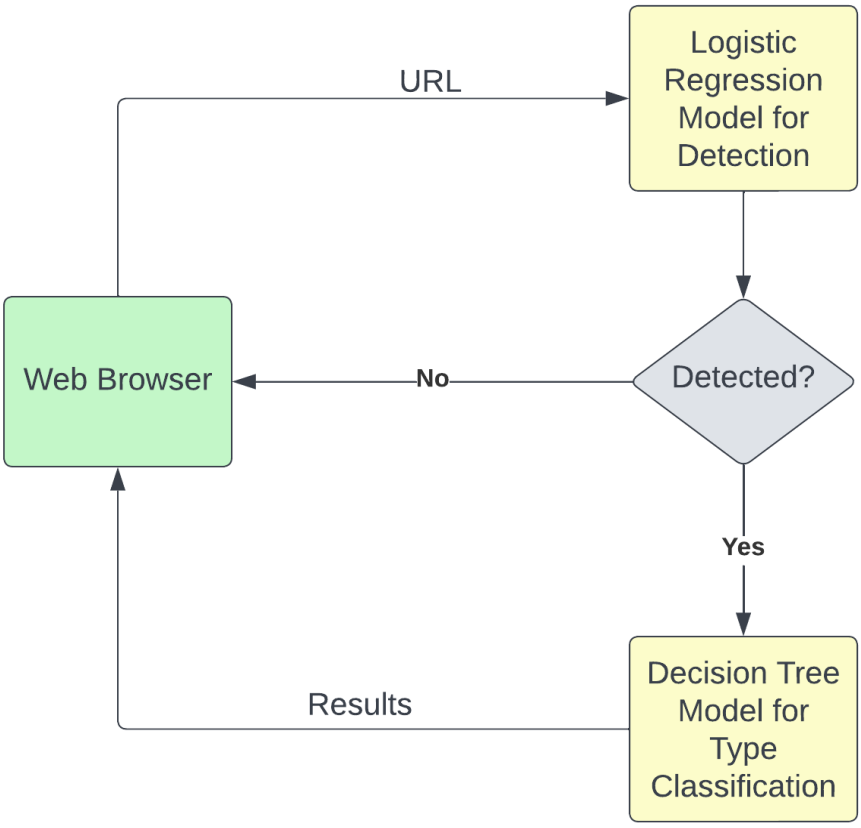
Security professionals and malware developers are constantly at odds with one another because malware sophistication grows as quickly as technology does. Modern cutting-edge research focuses on the development and application of machine learning algorithms in malware detection because they can keep up with the growth of malware. The study's goal is to provide a systematic and thorough analysis of deep learning-based machine learning techniques for malware detection. The following are the paper's main contributions:

1. It provides a comprehensive breakdown of the methods and elements used in a typical machine learning approach for locating and classifying malware.
2. It looks at the limitations and flaws of traditional machine learning.
3. It focuses on deep learning approaches while analyzing current market trends and technological breakthroughs.

The survey helps researchers better comprehend malware detection as a subject and the cutting-edge theories and approaches being employed by the academic community to address the issue [2].

Phishing attacks are the simplest method of obtaining sensitive information from unaware people. Phishers are eager in collecting sensitive information, such as usernames, passwords, and bank account details. Experts in cyber security are currently searching for dependable and consistent ways to identify phishing websites. The article employs machine learning to distinguish between legitimate and counterfeit URLs. It retrieves and analyzes different components from both sorts of URLs. Phishing websites can be recognized using algorithms like Decision Tree, Random Forest, and Support Vector Machine. In order to recognize phishing URLs and select the most efficient algorithm, the study evaluates the accuracy rates, false positive and false negative rates of various machine learning algorithms [3].

System Model



Proposed Work

We have made a Firefox Extension which when clicked, takes the URL of the page and sends it to the server via API. Server has both ML models file i.e., Logistic Regression for Detection and Decision Tree for Type Classification. Sever then feeds the input URL to the Detection Model, it will process it and will produce output as to whether any malicious behavior detected in the URL or not. If not detected, then the response will be reverted back to the browser stating the site is Safe to use. But if any malicious behavior is detected, then the URL will be given to the Decision Tree classifier for Type classification. The output of the classifier will be the type of the malicious behavior i.e., Benign, Defacement, Phishing or Malware.

Implementation Details

CVE Analyzer

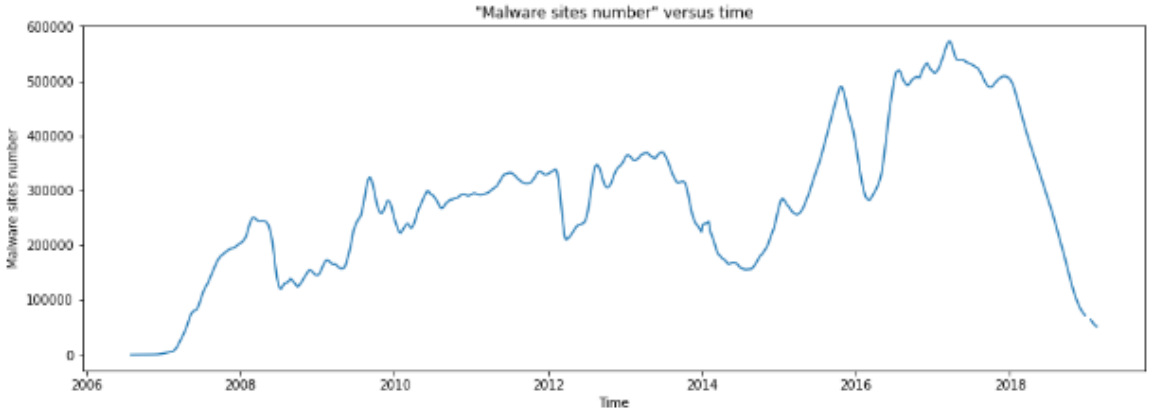


Figure 2: Malware Sites Count vs Time

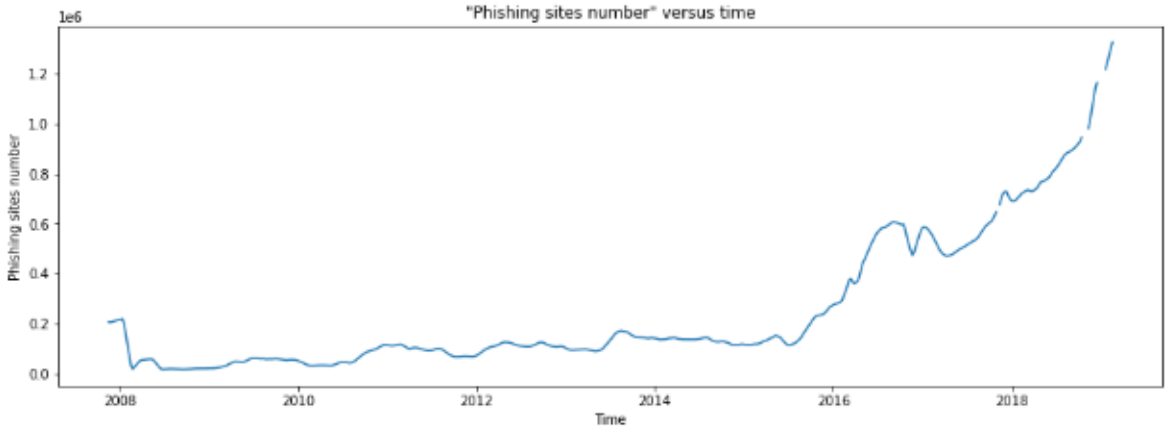


Figure 3: Phishing Sites Count vs Time

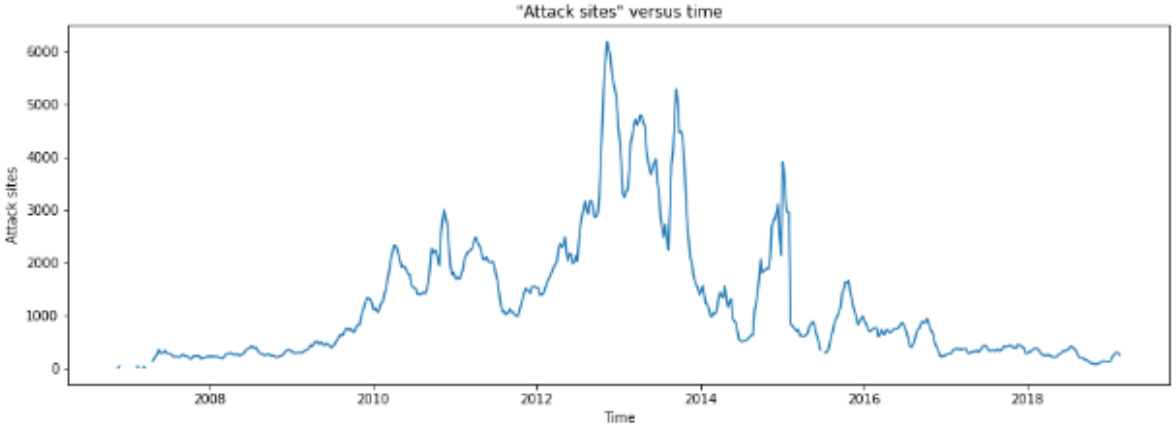


Figure 4: Attack Sites Count vs Time

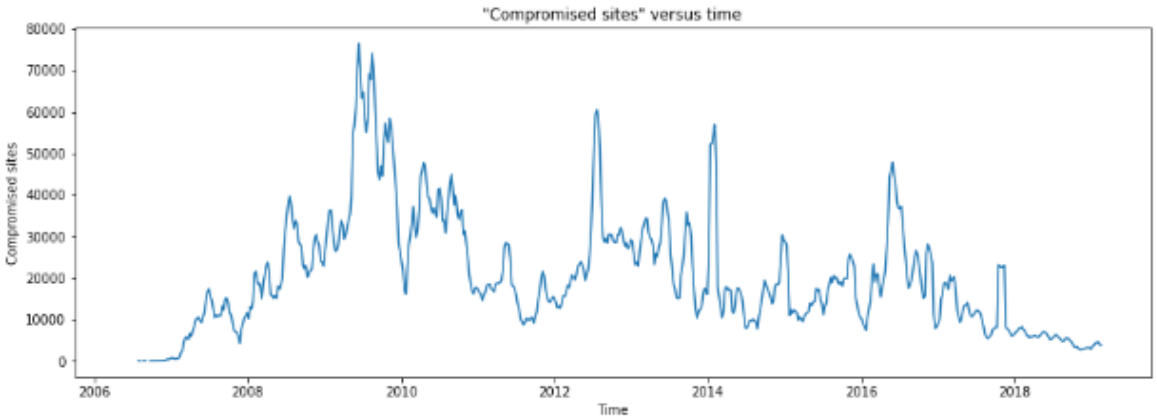


Figure 5: Compromised Sites Count vs Time

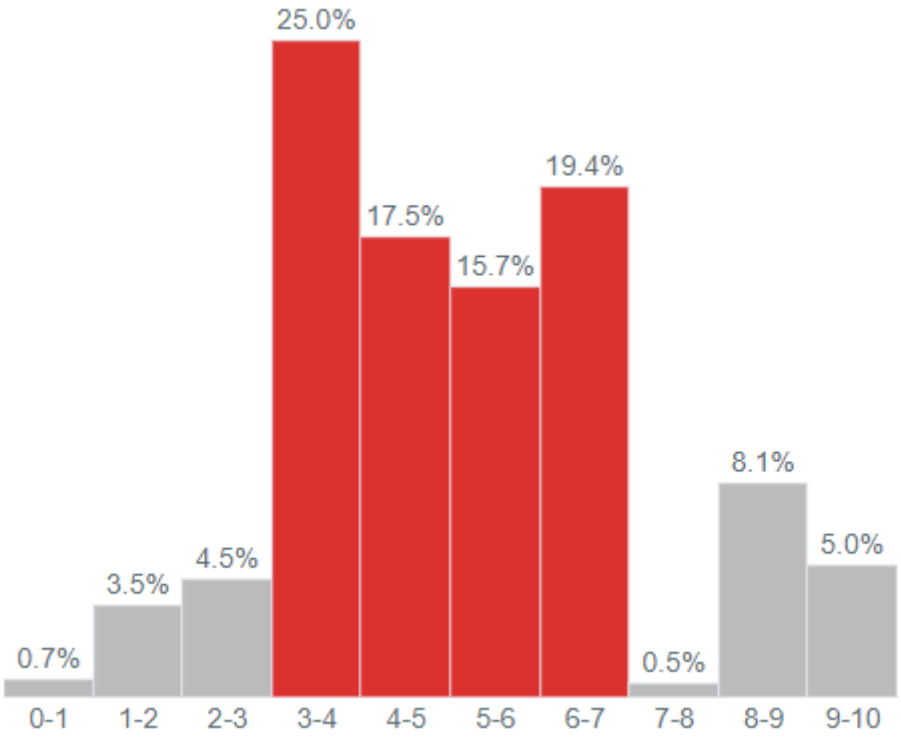


Figure 6: Threat Severity Distribution

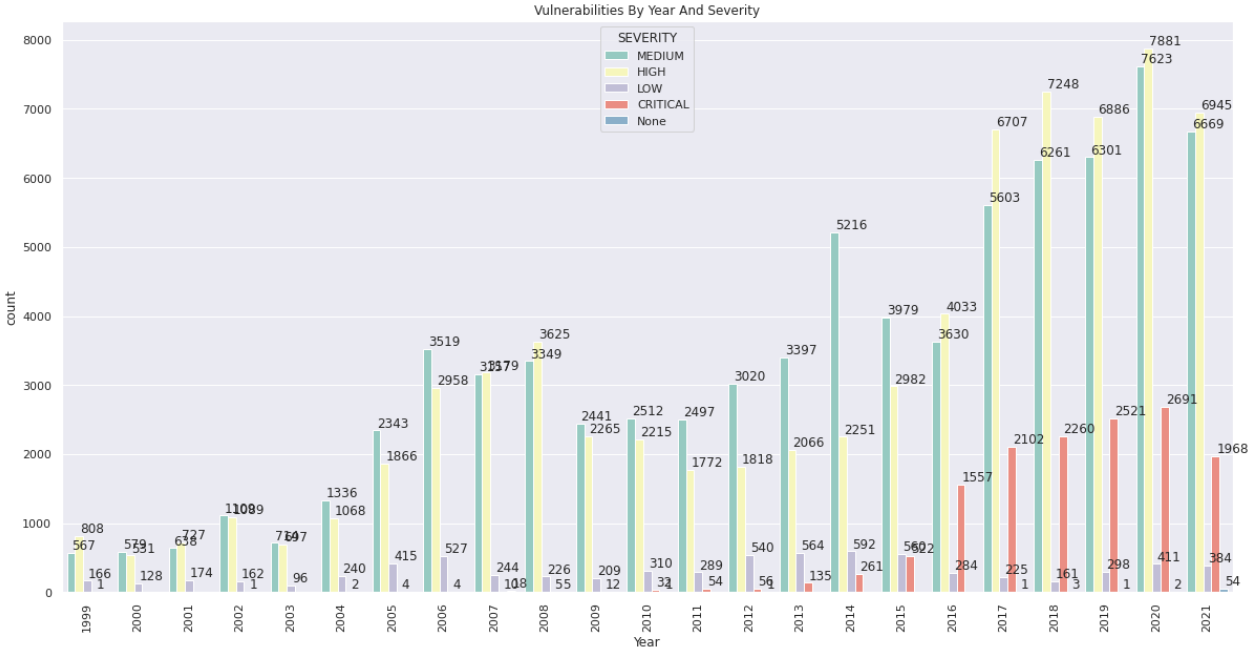


Figure 7: Vulnerabilities by Year and Severity

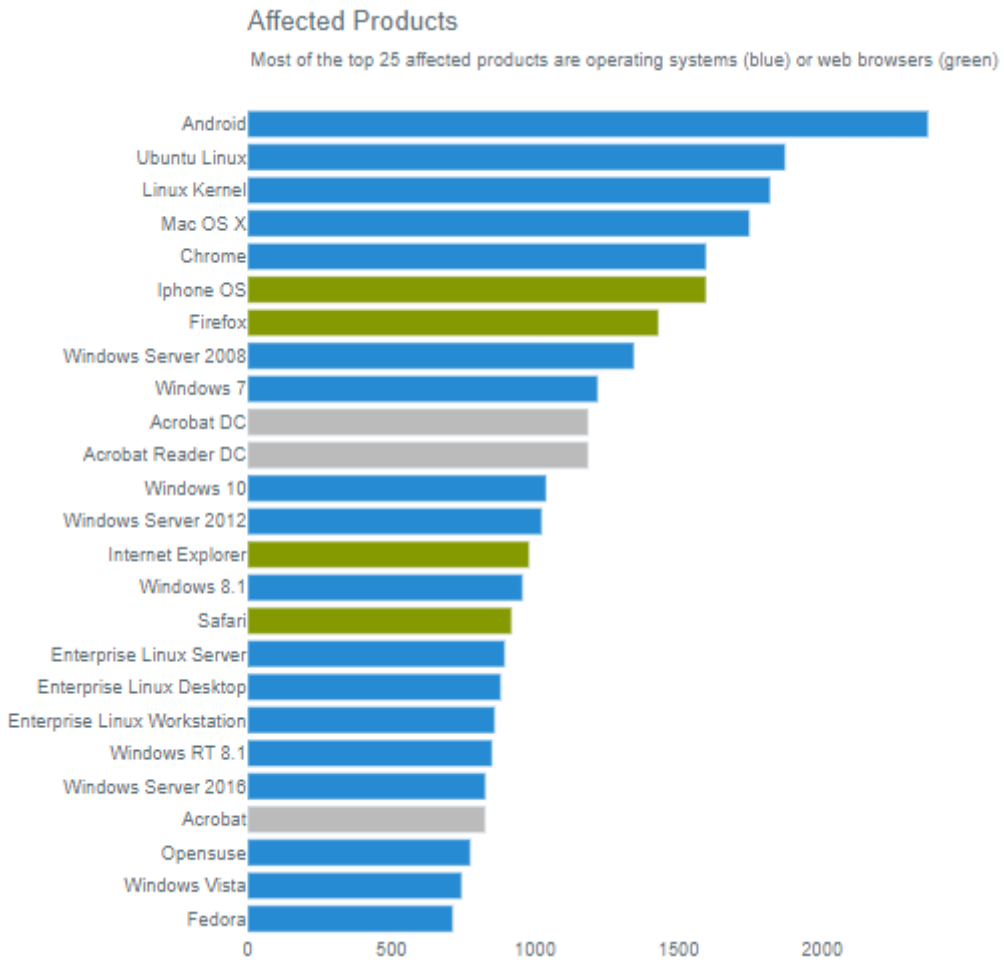


Figure 8: Affected Products

Detection part

1. Dataset:

In Dataset, number of records is 4,50,176 and number of columns is 3. Those columns are 'URL', 'label', 'result'. The types of the class are benign and malicious (phishing). Number of benign sites is 3,45,738 and the number of malicious is 1,04,438.

	url	label	result
0	https://www.google.com	benign	0
1	https://www.youtube.com	benign	0
2	https://www.facebook.com	benign	0
3	https://www.baidu.com	benign	0
4	https://www.wikipedia.org	benign	0

Figure 9: Head of the Dataset

	count	mean	std	min	25%	50%	75%	max
result	450176.0	0.231994	0.422105	0.0	0.0	0.0	0.0	1.0

Figure 10: Dataset Description

2. Data Preprocessing:

Dataset preprocessing depends on three features. The sites are given in 3 types of features and these features are further divided into sub-features.

Features are divided into 3 types:

A. Length Features.
B. Count Features.
C. Binary Features

A. Length Features:

It deals with the length specification of the URLs. Here we divide the URL in different types of lengths.

There are the types of additional features we took in consideration in the length features:

- Length Of URL.
- Length of Hostname.
- Length Of Path.
- Length Of First Directory.
- Length Of Top-Level Domain.

:

	url	label	result	url_length	hostname_length	path_length	fd_length	tld_length
0	https://www.google.com	benign	0	22	14	0	0	3
1	https://www.youtube.com	benign	0	23	15	0	0	3
2	https://www.facebook.com	benign	0	24	16	0	0	3
3	https://www.baidu.com	benign	0	21	13	0	0	3
4	https://www.wikipedia.org	benign	0	25	17	0	0	3

Figure 11: Dataset after length preprocessing

B. Count Features:

It deals with the counting of different characters specification of the URLs. Here we count the different types of characters like @, # etc. in the URL.

There are types of additional features we took in consideration in the count features:

- Count Of '-'
- Count Of '@'
- Count Of '?'
- Count Of '%'
- Count Of '.'
- Count Of '='
- Count Of 'http'
- Count Of 'www'
- Count Of Digits.
- Count Of Letters.
- Count Of Number of Directories.

	url	label	result	count-	count@	count?	count%	count.	count=	count-http	count-https	count-www	count-digits	count-letters	count_dir
0	https://www.google.com	benign	0	0	0	0	0	2	0	1	1	1	0	17	0
1	https://www.youtube.com	benign	0	0	0	0	0	2	0	1	1	1	0	18	0
2	https://www.facebook.com	benign	0	0	0	0	0	2	0	1	1	1	0	19	0
3	https://www.baidu.com	benign	0	0	0	0	0	2	0	1	1	1	0	16	0
4	https://www.wikipedia.org	benign	0	0	0	0	0	2	0	1	1	1	0	20	0

Figure 12: Dataset after count preprocessing

C. Binary Features:

It deals with the feature which can classify in yes or no. Here we consider the features like usage of IP, Shortening URL etc.

- Use of IP or not.
- Use of Shortening URL or not.

	url	label	result	short_url	use_of_ip
0	https://www.google.com	benign	0	1	1
1	https://www.youtube.com	benign	0	1	1
2	https://www.facebook.com	benign	0	1	1
3	https://www.baidu.com	benign	0	1	1
4	https://www.wikipedia.org	benign	0	1	1

Figure 13: Dataset after binary preprocessing

3. Data Visualization:

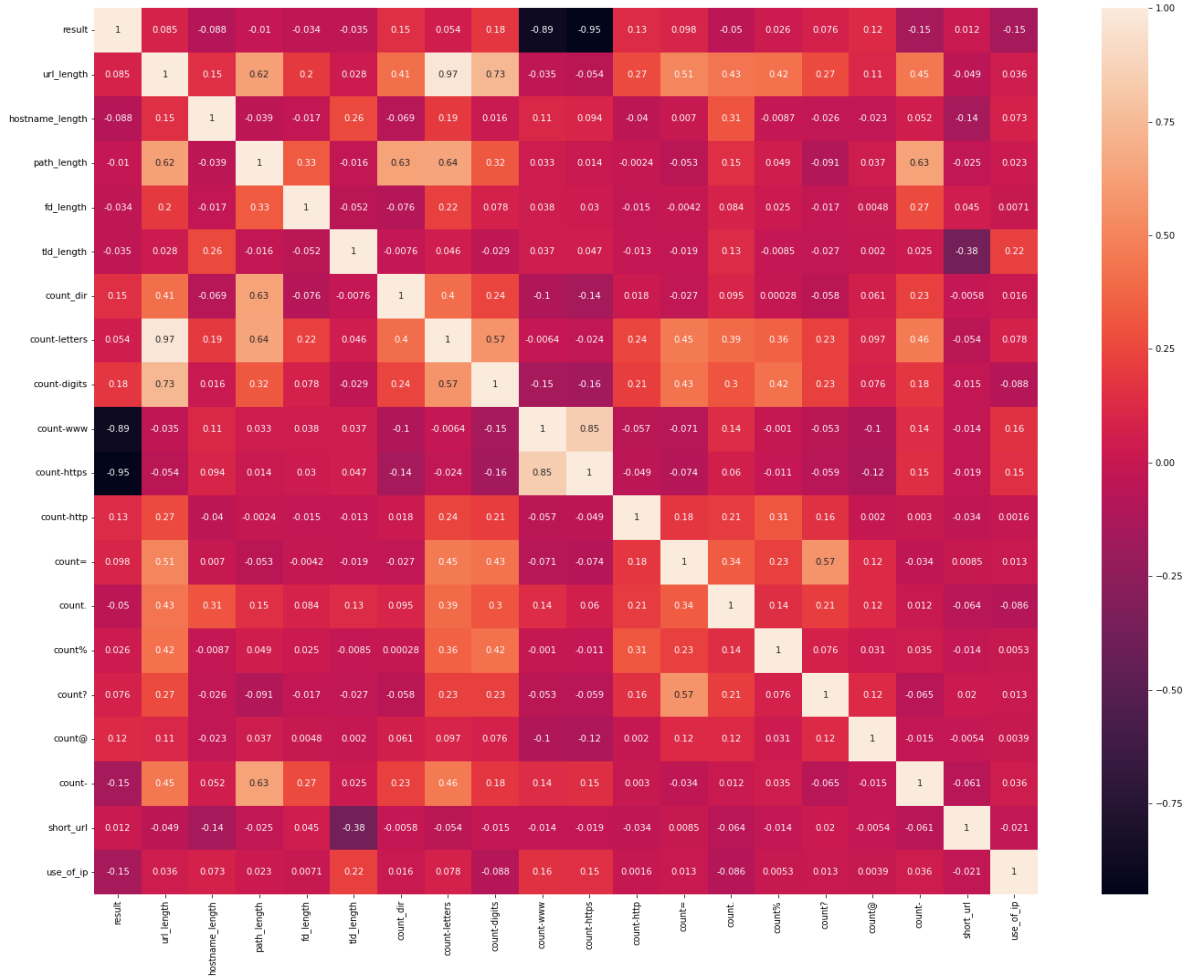


Figure 14: Heatmap between all features

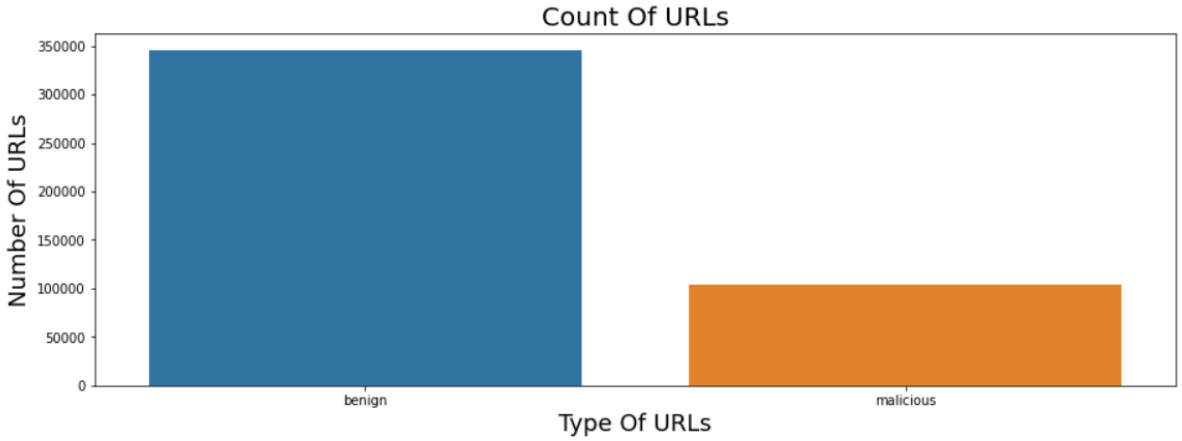


Figure 15: Count vs Type of URLs

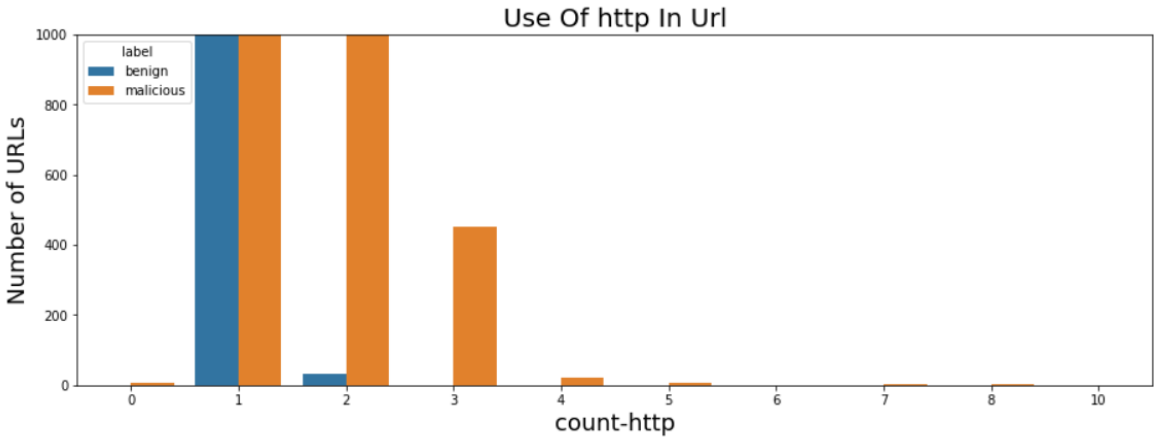


Figure 16: URL count vs HTTP count

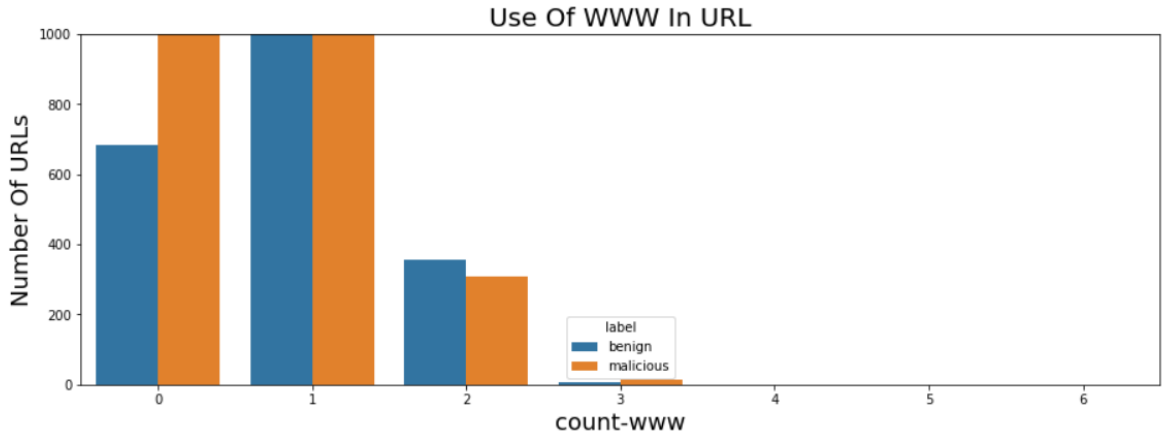


Figure 17: URL count vs WWW count

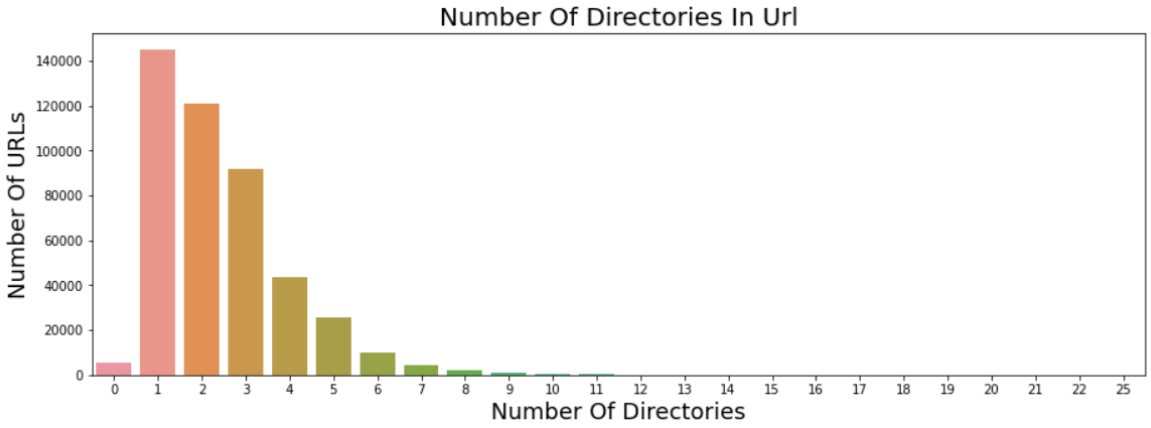


Figure 18: URL count vs Directories count

4. Model Building:

- We used the Train-Test Split ratio to be 70-30.
- The features that were taken into consideration while building models are: hostname_length, path_length, fd_length, tld_length, count-, count@, count?, count%, count., count=, count-http, count-https, count-www, count-digits, count-letters, count_dir, use_of_ip.

5. Classification Report:

I. Logistic Regression:

CLASSIFICATION REPORT

	precision	recall	f1-score	support
Bad	1.00	1.00	1.00	242154
Good	0.99	0.99	0.99	72970
accuracy			1.00	315124
macro avg	0.99	0.99	0.99	315124
weighted avg	1.00	1.00	1.00	315124

CONFUSION MATRIX

<AxesSubplot:>

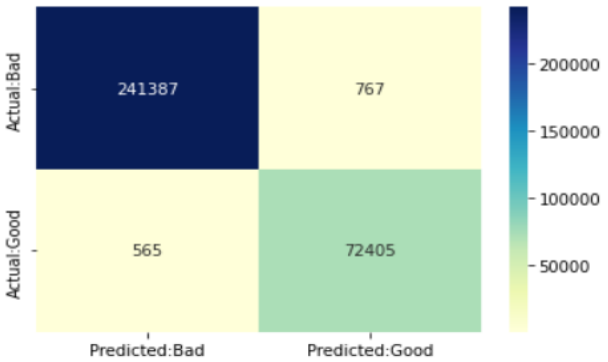


Figure 19: classification report & confusion matrix for logistic regression

II. Decision Tree:

CLASSIFICATION REPORT

	precision	recall	f1-score	support
Bad	1.00	1.00	1.00	241930
Good	0.99	0.99	0.99	73194
accuracy			1.00	315124
macro avg	0.99	0.99	0.99	315124
weighted avg	1.00	1.00	1.00	315124

CONFUSION MATRIX

<AxesSubplot:>

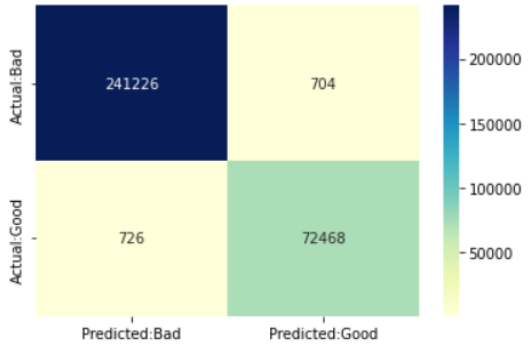


Figure 20: classification report & confusion matrix for decision tree

III. Random Forest:

CLASSIFICATION REPORT

	precision	recall	f1-score	support
Bad	1.00	1.00	1.00	242268
Good	0.99	1.00	0.99	72856
accuracy			1.00	315124
macro avg	1.00	1.00	1.00	315124
weighted avg	1.00	1.00	1.00	315124

CONFUSION MATRIX

<AxesSubplot:>

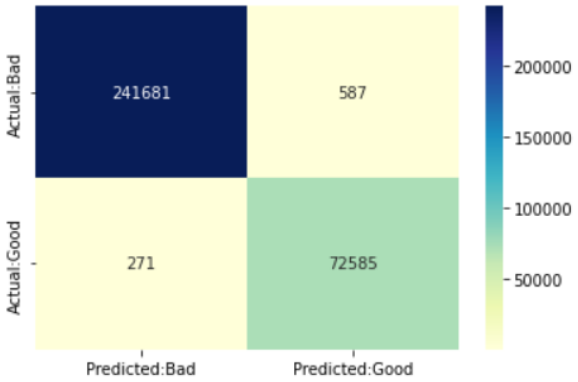


Figure 21: classification report & confusion matrix for random forest

IV. Support Vector Classifier:

CLASSIFICATION REPORT

	precision	recall	f1-score	support
Bad	1.00	1.00	1.00	242000
Good	0.99	0.99	0.99	73124
accuracy			1.00	315124
macro avg	1.00	1.00	1.00	315124
weighted avg	1.00	1.00	1.00	315124

CONFUSION MATRIX

<AxesSubplot:>

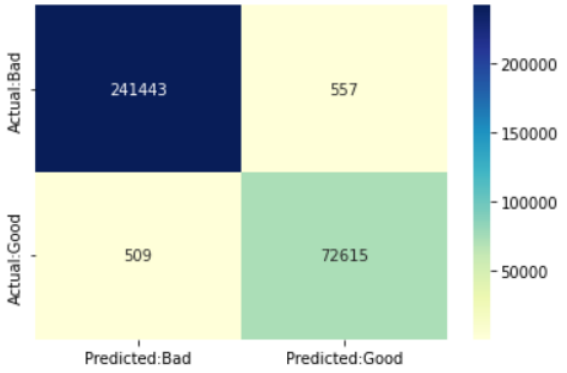


Figure 22: classification report & confusion matrix for support vector classifier

V. Gaussian Naïve Bayes Classifier:

CLASSIFICATION REPORT

	precision	recall	f1-score	support
Bad	0.99	1.00	0.99	240203
Good	0.99	0.97	0.98	74921
accuracy			0.99	315124
macro avg	0.99	0.98	0.99	315124
weighted avg	0.99	0.99	0.99	315124

CONFUSION MATRIX

<AxesSubplot:>

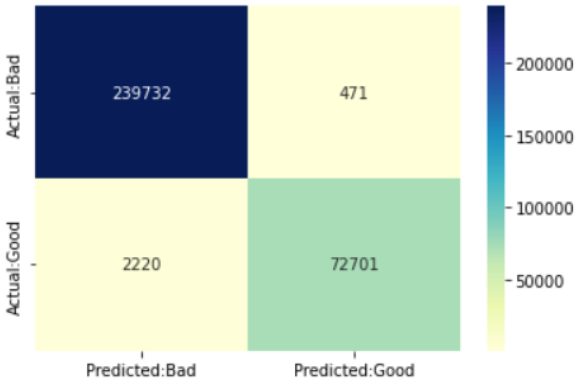


Figure 23: classification report & confusion matrix for gaussian naive bayes classifier

VI. Prediction:

```
# link = input("Enter your link:\t")

link = "www.youtube.com"
pred(link)
```

```
benign
```

+ Code + Markdown

```
link = "svision-online.de/mgfi/administrator/components/com_babackup/classes/fx29id1.txt"
pred(link)
```

```
Phishing
```

Figure 23: prediction of different URLs from model

Classification Part

1. Dataset:

6,51,191 URLs total, including 4,28,103 benign or safe URLs, 96,457 defacement URLs, 94,111 phishing URLs, and 32,520 malware URLs, have been gathered into a sizable dataset by our team. As we all know, selecting the right dataset for a machine learning project is one of the most important responsibilities. This dataset is a compilation from five sources.

We collected benign, phishing, malware, and defacement URLs using URL dataset (ISCX-URL-2016) We have observed an uptick in phishing and malware URLs using the information from the Malware domain black list. We increased the variety of benign URLs by utilizing the Faizan git repository. Finally, we have increased the number of phishing URLs by using the Phishtank dataset and the PhishStorm dataset.html. The dataset, as we already indicated, was assembled from a variety of sources. In order to preserve only the URLs and their class type, we first collected the URLs from several sources into a separate data frame.

	url	type
0	br-icloud.com.br	phishing
1	mp3raid.com/music/krizz_kaliko.html	benign
2	bopsecrets.org/rexroth/cr/1.htm	benign
3	http://www.garage-pirenne.be/index.php?option=...	defacement
4	http://adventure-nicaragua.net/index.php?optio...	defacement
5	http://buzzfil.net/m/show-art/ils-etaient-loin...	benign
6	espn.go.com/nba/player/_/id/3457/brandon-rush	benign
7	yourbittorrent.com/?q=anthony-hamilton-soulife	benign
8	http://www.pashminaonline.com/pure-pashminas	defacement
9	allmusic.com/album/crazy-from-the-heat-r16990	benign
10	corporationwiki.com/Ohio/Columbus/frank-s-bens...	benign
11	http://www.ikenmijnkunst.nl/index.php/expositi...	defacement
12	myspace.com/video/vid/30602581	benign
13	http://www.lebensmittel-ueberwachung.de/index....	defacement
14	http://www.szabadmunkaero.hu/cimoldal.html?sta...	defacement
15	http://larcadelcarnevale.com/catalogo/palloncini	defacement
16	quickfacts.census.gov/qfd/maps/iowa_map.html	benign
17	nugget.ca/ArticleDisplay.aspx?archive=true&e=1...	benign
18	uk.linkedin.com/pub/steve-rubenstein/8/718/755	benign
19	http://www.vnic.co/khach-hang.html	defacement

Figure 24: Head of the Dataset

Defacements refer to unauthorized additions, omissions, or modifications to web pages that include material. Hacktivists often carry out these attacks by infiltrating a website or web server and changing or replacing the hosted website's content with their own messaging.

Phishing is a form of social engineering assault that is frequently used to obtain user information, such as login credentials and credit card numbers. It occurs when an offender persuades a victim to open an email, instant message, or text message by assuming the identity of a reliable source.

Malware, short for "malicious software," is a term for a file or piece of code that may essentially perform whatever action a hacker desires, including infecting, investigating, stealing, and performing operations. There are many different ways to infect computers because viruses come in such a wide variety.

```
benign         428103
defacement      96457
phishing        94111
malware         32520
Name: type, dtype: int64
```

Figure 25: Count of Types

2. Data Preprocessing:

All the detection features are considered and along with it we also considered extra three features and those are:
Abnormal_url, count of digits, count of all letters.

3. Data Visualization:

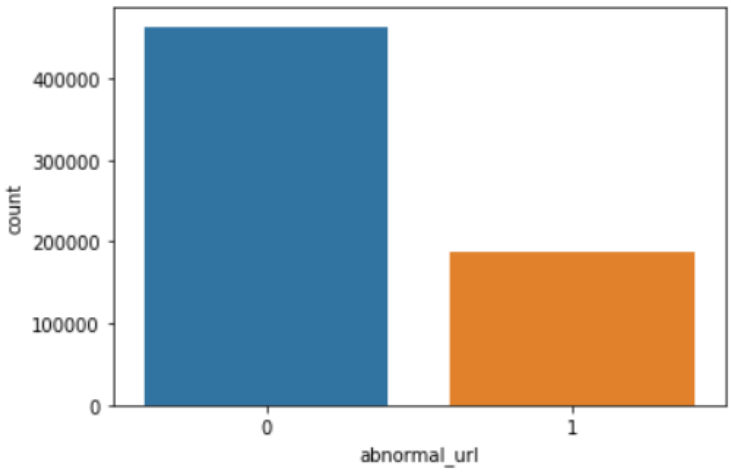


Figure 26: URL count vs abnormal_url count

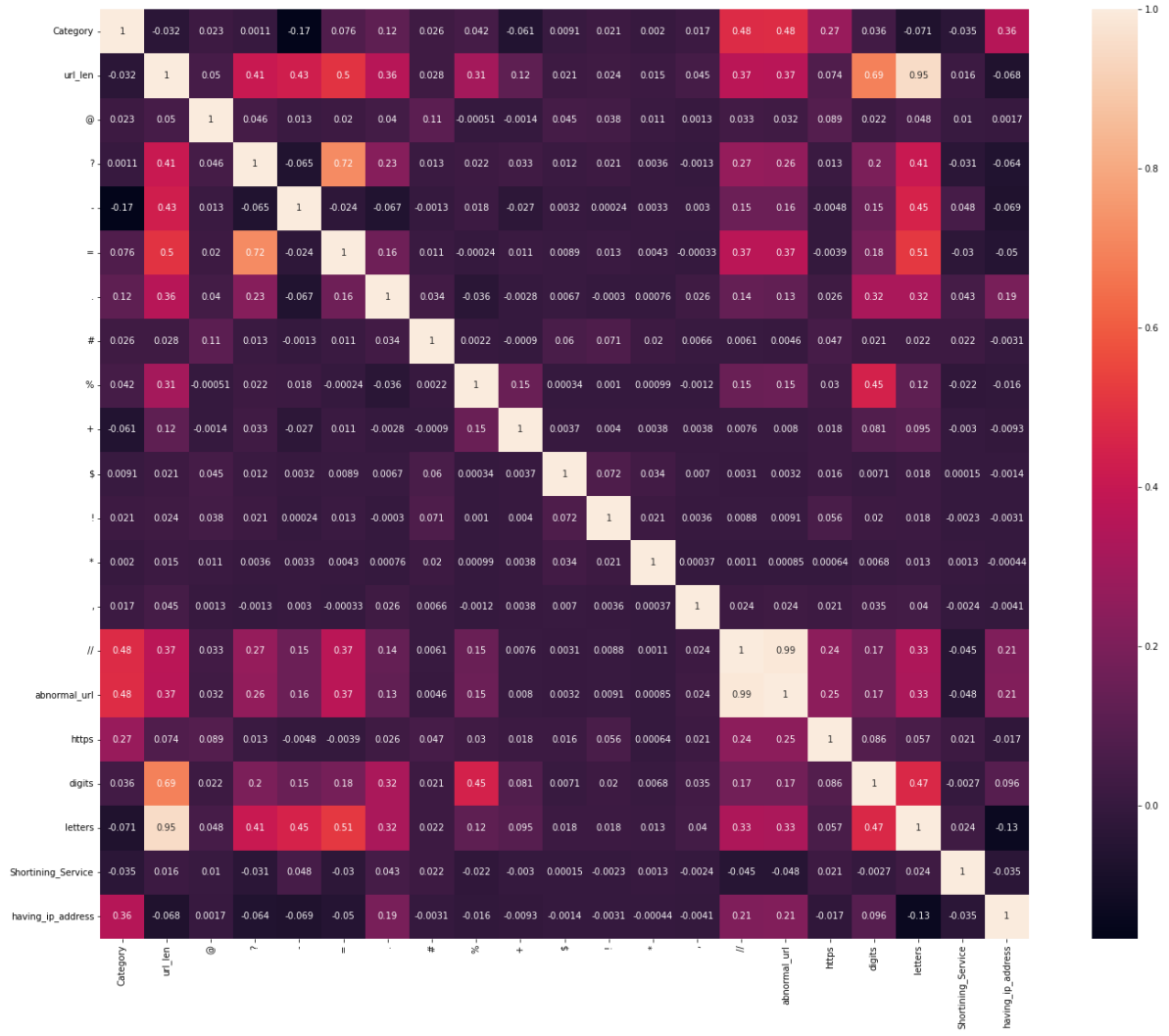


Figure 27: Heatmap between features

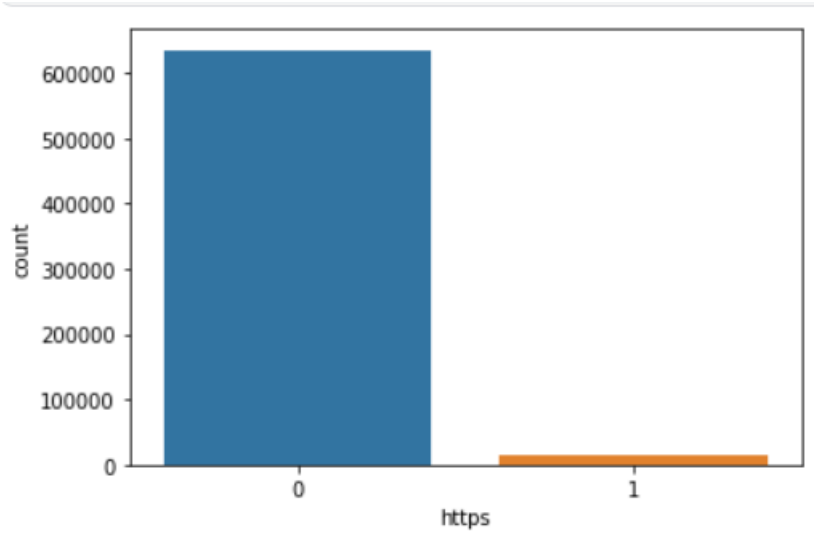


Figure 28: URL count vs HTTPS count

4. Model Building:

- We used the Train-Test Split ratio to be 80-20.
- All the features that were there in detection part were taken into consideration while building models along with features like:
  Abnormal_url, count of digits, count of all letters.

5. Classification Report:

I. Logistic Regression:

Classification_report

	precision	recall	f1-score	support
0	0.83	0.97	0.90	85565
1	0.75	0.83	0.79	19319
2	0.30	0.07	0.12	18805
3	0.71	0.53	0.61	6550
accuracy			0.80	130239
macro avg	0.65	0.60	0.60	130239
weighted avg	0.74	0.80	0.75	130239

Figure 29: classification report & confusion matrix for logistic regression

II. Decision Tree:

Classification_report

	precision	recall	f1-score	support
0	0.92	0.97	0.94	85565
1	0.93	0.96	0.95	19319
2	0.80	0.57	0.66	18805
3	0.95	0.91	0.93	6550
accuracy			0.91	130239
macro avg	0.90	0.85	0.87	130239
weighted avg	0.90	0.91	0.90	130239

Confusion_matrix

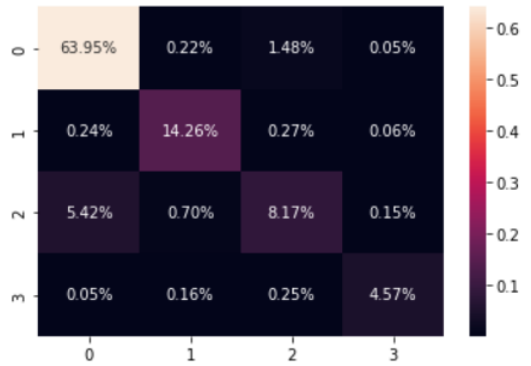


Figure 30: classification report & confusion matrix for decision tree

III. Random Forest:

Classification_report

	precision	recall	f1-score	support
0	0.92	0.98	0.95	85565
1	0.94	0.97	0.95	19319
2	0.83	0.58	0.68	18805
3	0.96	0.91	0.93	6550
accuracy			0.91	130239
macro avg	0.91	0.86	0.88	130239
weighted avg	0.91	0.91	0.91	130239

Confusion_matrix

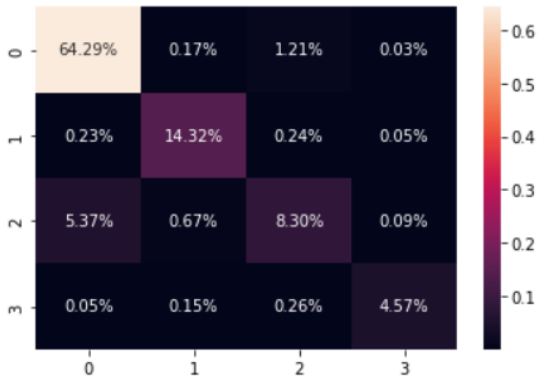


Figure 31: classification report & confusion matrix for random forest

IV. Gaussian Naïve Bayes Classifier:

Classification_report

	precision	recall	f1-score	support
0	0.85	0.92	0.88	85565
1	0.66	1.00	0.79	19319
2	0.60	0.02	0.04	18805
3	0.61	0.70	0.65	6550
accuracy			0.79	130239
macro avg	0.68	0.66	0.59	130239
weighted avg	0.77	0.79	0.74	130239

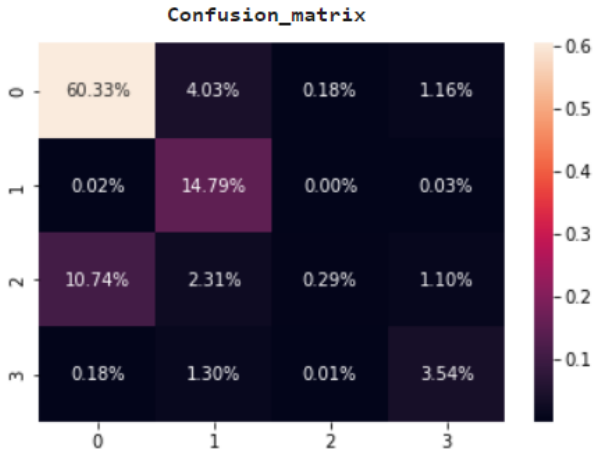


Figure 32: classification report & confusion matrix for Gaussian Naïve Bayes Classifier

6. Prediction:

```
# link = input("Enter your link:\t")

link = "www.youtube.com"
pred(link)
```

```
benign
```

+ Code + Markdown

```
link = "svision-online.de/mgfi/administrator/components/com_babackup/classes/fx29id1.txt"
pred(link)
```

```
Phishing
```

```
links1 = 'vanderbilt.rivals.com/viewcoach.asp?coach=2079&sport=1&year=2011'
pred(links1)
```

```
Benign
```

```
link2 = 'http://www.kingsmillshotel.com/spring/mothers-day'
pred(link2)
```

```
Defacement
```

```
link3 = 'https://docs.google.com/spreadsheet/viewform?formkey=dGg2Z1lCUH1SdjllTVNRUW50TFIzSkE6MQ'
pred(link3)
```

```
Phishing
```

```
link4 = 'http://9779.info/%E6%A0%91%E5%8F%B6%E7%B2%98%E8%B4%B4%E7%94%BB/'
pred(link4)
```

```
Malware
```

Figure 33: prediction of different URLs from model

Results and comparison with the existing work

Detection

Sr. no.	Model	Training Accuracy (%)	10-fold Cross Validation mean accuracy (%)	Testing accuracy (%)	Time Taken (sec)
1.	Logistic Regression	99.5957	99.5957	99.5773	2.4
2.	Decision Tree	99.9518	99.9529	99.5462	1
3.	Random Forest	99.9518	99.7223	99.7277	21.1
4.	Support Vector Classifier	99.6630	99.6586	99.6617	262
5.	Gaussian Naive Bayes classifier	99.1469	99.1669	99.1460	1.3

Table 1: Accuracies and time taken of different models for detection

Classification

Sr. no.	Model	Training Accuracy (%)	10-fold Cross Validation mean accuracy (%)	Testing accuracy (%)	Time Taken (sec)
1.	Logistic Regression	79.69	79.22	79.76	78.066
2.	Decision Tree	93.60	90.65	90.94	5.92
3.	Random Forest	93.59	91.26	91.47	130
4.	Gaussian Naive Bayes classifier	78.96	78.96	78.95	1.4

Table 2: Accuracies and time taken of different models for classification

SUMMARY AND FUTURE DIRECTIONS

This work aims to improve the identification and classification of malicious websites through the use of machine learning techniques. For detection part, we achieved highest testing accuracy in Random Forest i.e., 99.7% but our simple logistic regression is giving testing accuracy 99.6%, which is way less complex in terms of time and resources. So, for detection part we choose logistic regression. For classification part, we achieved highest testing accuracy in Random Forest i.e., 91.47% but again decision tree is giving accuracy nearly 91% which is way less complex in terms of time and resources. So, for classification part we choose decision tree.

The results also demonstrate that classifier performance rises as training data volume increases. To more accurately identify phishing websites in the future, hybrid technology that combines the logistic regression & decision tree algorithm of machine learning technology with the blacklist technique will be used. The CVE Analyzer can be improved more in future from which we can get more clear understanding of the situation. The system's usefulness can be improved by utilizing other cutting-edge learning approaches, such as deep learning, even though the detection rate results produced are satisfactory. We also have access to a huge dataset. Any dataset can be used to build the knowledge base when using this deep learning technique. However, more training time is required for this. The system can therefore support some parallel processing techniques.

It may also be possible to develop a separate subsystem for shorter URLs that merely contain a domain or subdomain. Some recent information about the website, such as the recent week’s visitor’s number, date when the domain name was registered, etc., is required to detect these types of web sites. These traits might not be selected for real-time detection because of the additional time needed to check them. The network may put a website on its local blacklist and forbid it from accepting any more requests if it is later determined to be a phishing site.