PHISHING WEBSITE DETECTION USING MACHINE LEARNING

A PROJECT REPORT

Submitted by

Shiv Sai, 19BCT0008

Pranav D S, 19BCT0135

Course Code: CSE3501

Course Title: Information Security Analysis and Audit

Under the guidance of

Dr. Kakelli Anil Kumar

Associate Professor

SCOPE, VIT, Vellore.

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

December -2021

INDEX

1. INTRODUCTION

A phishing website is a social engineering method that replicates trustful uniform resource locators (URLs) and webpages. It's used to exploit sensitive user data that includes login credentials and credit card numbers. A person is prone to this attack when an attacker, disguised as a trusted organization, tricks the user into opening an email or a text message. Machine learning is a method of data analysis that is used to create AI models. It is a branch of artificial intelligence where systems learn from data, identify patterns and make decisions with minimal human intervention. The objective of this project is to train machine learning models on a dataset created to predict phishing websites and to prevent users from getting attacked.

To develop this project, we will be using Jupyter Notebook and Python. The dataset consisting of phishing URLs is taken from an open-source service called Phish Tank. This service provides a set of phishing URLs in multiple formats like csv, json etc. that gets updated hourly.

2. LITERATURE SURVEY

2.1 Detecting Phishing Websites by Looking at Them

A phishing detection approach called Phish Zoo that uses profiles of trusted websites' appearances to detect phishing. This was authored by Sadia Afroz and Rachel Greenstadt. Profile matching approach depends only on the current contents of a site. This approach can provide user-customized phishing protection. However, fails to find matching in cases where the logos are rotated more than 30 degrees.

2.2 Detecting Phishing Websites through Deep Reinforcement Learning

The proposed model is capable of adapting to the dynamic behavior of the phishing websites and thus learning the features associated with phishing website detection.This paper was authored by Moitrayee Chatterjee and Akbar Siami Namin.This deep learning model is a dynamic and self-adaptive phishing identification framework. This work is however not optimized for real world implementation.

2.3 Heuristic nonlinear regression strategy for detecting phishing websites

This paper was authored by Mehdi Babagoli, Mohammad, Pourmahmood, Aghababa and Vahid Solouk. This research utilizes feature selection methods to select the best feature subset and then apply meta-heuristic algorithms to detect the phishing websites. The proposed method for phishing detection has high degree of efficiency than some of the conventional mentioned methods. Applying the meta-heuristic algorithm in phishing detection methods has not been analyzed yet.

2.4 An efficacious method for detecting phishing webpages through target domain identification

This was authored by Gowtham Ramesh, Ilango Krishnamurthi and K Sampath Sree Kumar. The author has proposed an anti-phishing technique that groups the domains from hyperlinks having direct or indirect association with the given suspicious webpage. The proposed method has the advantage of detecting phishing webpages of any language. In this approach we cannot detect phishing webpages hosted on the compromised domains.

2.5 Detecting Phishing Websites using Automation of Human Behavior

This was authored by Routhu Srinivasa Rao and Alwyn R Pais. The application Feedphish, logs into a website using fake values. If the web page logs in successfully, it is classified as phishing otherwise it undergoes further heuristic filtering. The proposed application neither demands third party services nor prior knowledge like web history, whitelist or blacklist of URLS. The application is able to deal login windows which contains anchor text like sign in, signin, login, log in, but it fails to handle if the anchor text is replaced with an icon or an image.

2.6 Phishing Detection: Analysis of Visual Similarity Based Approaches. Security and Communication Network

This paper was authored by Jain A.K. and Gupta B.B. This paper focusses on visual similarity based phishing detection techniques utilise the feature set like text content, text format, HTML tags, CSS, image, and so forth, to make the decision. This paper presents a comprehensive analysis of phishing attacks, their exploitation. If the phishing website is partially copied (less than 50%) from the legitimate website, then none of the visual similarity based approaches can detect it.

2.7 Towards Lightweight URL-Based Phishing Detection

It was authored by Butnaru, Mylonas and Pitropakis N. This paper uses supervised machine learning to block phishing attacks, based on a novel combination of features that are extracted solely from the URL. It evaluates performance over time with a dataset which consists of active phishing attacks and compare it with Google Safe Browsing (GSB). This paper proposes and evaluates a phishing detection engine, which uses supervised machine learning in order to detect phishing attacks based on novel combination features that are extracted from the URL. The paper has to use multiple latest data sets to verify its capabilities.

2.8 Phishing Detection: A Literature Survey

This paper was authored by Mahmoud Khonji, Youssef Iraqi, Senior Member, IEEE, and Andrew Jones**.** This is a theoretical Paper on Phishing and its detection. ML classifiers can automatically evolve via reinforcement learning. It is also possible to periodically construct newer classification models by simply retraining the learner with updated sample data sets. Phishing detection techniques from the perspective of their computational cost and energy consumption is very high.

2.9 Learning to Detect Phishing Emails

This paper was authored by Ian Fette, Norman Sadeh and Anthony Tomasic. This paper uses a method for detecting these attacks, by an application of machine learning on a feature set designed to highlight user-targeted deception in emails. The approach used is flexible, and new external information sources can be added as they become available. Lower accuracy when the model also evaluates spam folder.

2.10 DeltaPhish: Detecting Phishing Webpages in Compromised Websites

Authored by Igino Corona Battista Biggio ,Matteo Contini, Luca Piras, Roberto Corda ,Mauro Mereu ,Guido Mureddu, Davide Ariu and Fabio Roli.. Phishing webpages can be accurately detected by highlighting HTML code and visual differences with respect to other legitimate pages hosted within a compromised website. The approach is also robust to well-crafted manipulations of the HTML code of the phishing page to evade detection. It is not able to detect phishing pages hosted through other means than compromised websites.

3. OVERVIEW OF THE WORK

We have developed a Machine Learning model that can detect if a given website's URL (Uniform Resource Locator) is phishing or a legitimate site. The model works based on identifying features of a website that can be used to detected if the site is a phishing website. For example, an IP address in the Domain name or the presence of '//' in the website (meaning redirection to other pages once the URL is clicked) are some characteristics of phishing websites.

3.1 PROBLEM DESCRIPTION

Phishing is one of the major threats in this internet era. Phishing is a smart process where a legitimate website is cloned and victims are lured to the fake website to provide their personal as well as confidential information, sometimes it proves to be costly. Though most of the websites will give a disclaimer warning to the users about phishing, users tend to neglect it. It is not a fully responsible action by the websites also and there is not much that the websites could really do about it.

3.2 WORKING MODEL

There are a lot of algorithms and a wide variety of data types for phishing detection in the academic literature and commercial products. A phishing URL and the corresponding page have several features which can be differentiated from a malicious URL. For example; an attacker can register long and confusing domain to hide the actual domain name (Cybersquatting, Typo squatting). In some cases, attackers can use direct IP addresses instead of using the domain name. This type of event is out of our scope, but it can be used for the same purpose. Attackers can also use short domain names which are irrelevant to legitimate brand names and don't have any Free URL addition. But these type of web sites are also out of our scope, because they are more relevant to fraudulent domains instead of phishing domains.

Beside URL-Based Features, different kinds of features which are used in machine learning algorithms in the detection process of academic studies are used. Features collected from academic studies for the phishing domain detection with machine learning techniques are grouped as given below.

1. URL-Based Features
2. Domain-Based Features

3. Page-Based Features

4. Content-Based Features

Our model focuses on URL-based features for identification of malicious features and classification of sites into phishing and legitimate.

3.3 DESIGN DESCRIPTION

URL is the first thing to analyse a website to decide whether it is a phishing or not. As we mentioned before, URLs of phishing domains have some distinctive points. Features which are related to these points are obtained when the URL is processed. Some of URL-Based Features are given below.

1.Digit count in the URL

2.Total length of URL

3.Checking whether the URL is Typosquatted or not. (google.com → goggle.com)

4.Checking whether it includes a legitimate brand name or not (apple-icloud-login.com)

5. Number of subdomains in URL

6. Is Top Level Domain (TLD) one of the commonly used one?

Page-Based Features:

Page-Based Features are using information about pages which are calculated reputation ranking services. Some of these features give information about how much reliable a web site is. Some of Page-Based Features are given below.

1. Global Page rank

2.Country Page rank

3.Position at the Alexa Top 1 million Site There so many machine learning algorithms and each algorithm has its own working mechanism.

In this project, we have explained Decision Tree Algorithm and Random Forest Classifier to develop our machine learning models and we have tested the accuracy of both these models on our dataset.

4. IMPLEMENTATION

We have taken two different datasets one for phishing URL's and the other for legitimate URL's and combined them into a single dataset. The dataset consisting of phishing URLs is taken from an open-source service called PhishTank. This service provides a set of phishing URLs in multiple formats like csv, json etc. that gets updated hourly. The model is developed on Jupyter Notebook using Python. Data Preprocessing was done using Pandas. Plots and graphs were developed using Matplotlib. The model was developed using two algorithms – Decision Tree and Random Forest, the accuracy of both these algorithms were also compared. Our project is divided into three modules

1. Preprocessing
2. Feature Extraction
3. Training and Testing

4.1 PREPROCESSING

Preprocessing was done to both datasets separately. This first step in preprocessing was to split the protocol (Example: http://) from the rest of the URL. The string before '//' contained the protocol and after it contained the domain name and address. Next step was to spilt the domain name and the path of the URL. This was done by splitting the string at the first '/'. The part before '/' is the domain name and after it was the address/path of the URL. This splitting was saved as a data frame and then sent to Feature Extraction.

4.2 FEATURE EXTRACTION

The second module was Feature extraction. In feature extraction, we looked for parts of the URL's that could hint the URL being a phishing site. The list of features that hint the website link to possibly be a phishing site was taken from the research journal "Phishing Websites Features" by Rami M. Mohammad, Fadi Thabtah and Lee McCluskey. The following features were considered for classification of websites into phishing and legitimate.

4.2.1 URL Length

Phishers can use long URL to hide the doubtful part in the address bar. For example: http://federmacedoadv.com.br/3f/aze/ab51e2e319e51502f416dbe46b773a5e/?cmd=_home&dispatch=11004d58f5b74f8dc1e7c2e8dd4105e811004d58f5b74f8dc1e7c2e8dd4105e8@phishing.website.html. If the length of the URL is less than 54, it is considered as legitimate, if it is between 54 and 75, we classify it as suspicious and if the length is greater than or equal 75 characters then the URL classified as phishing.

4.2.2 URL's having @ symbol

Using "@" symbol in the URL leads the browser to ignore everything preceding the "@" symbol and the real address often follows the "@" symbol. Therefore, if @ symbol is present, we classify it as phishing.

4.2.3 Redirecting using "//"

The existence of "//" within the URL path means that the user will be redirected to another website. An example of such URL's is: "http://www.legitimate.com//http://www.phishing.com." We examine the location where the "//" appears. After we split the URL into protocol, domain name and address in preprocessing. The presence of a '//' in the address part will indicate that redirection is being used. Such URL's as classified as phishing.

4.2.4 Adding Prefix or Suffix Separated by (-) to the Domain

The dash symbol is rarely used in legitimate URLs. Phishers tend to add prefixes or suffixes separated by (-) to the domain name so that users feel that they are dealing with a legitimate webpage. For example, http://www.Confirme-paypal.com/. Therefore, we mark websites with the presence of "-" in the domain name as Phishing.

4.2.5 Sub-Domain and Multi Sub-Domains

Let us assume we have the following link: http://www.hud.ac.uk/students/. A domain name might include the country-code top-level domains (ccTLD), which in our example is “uk”. The “ac” part is shorthand for “academic”, the combined “ac.uk” is called a second-level domain (SLD) and “hud” is the actual name of the domain. To produce a rule for extracting this feature, we firstly have to omit the (www.) from the URL which is in fact a sub domain in itself. Then, we have to remove the (ccTLD) if it exists. Finally, we count the remaining dots. If the number of dots is greater than one, then the URL is classified as “Suspicious” since it has one sub domain. However, if the dots are greater than two, it is classified as “Phishing” since it will have multiple sub domains. Otherwise, if the URL has no sub domains, we will assign “Legitimate” to the feature.

4.2.6 Using the IP Address

If an IP address is used as an alternative of the domain name in the URL, such as “http://125.98.3.123/fake.html”, users can be sure that someone is trying to steal their personal information. Sometimes, the IP address is even transformed into hexadecimal code as shown in the following link “http://0x58.0xCC.0xCA.0x62/2/paypal.ca/index.html”.

4.2.7 URL Shortening Services

URL shortening is a method on the “World Wide Web” in which a URL may be made considerably smaller in length and still lead to the required webpage. This is accomplished by means of an “HTTP Redirect” on a domain name that is short, which links to the webpage that has a long URL. For example, the URL “http://portal.hud.ac.uk/” can be shortened to “bit.ly/19DXSk4”. In our model, we have stored a list of shortening service URLs to verify if the website is using one of them.

4.2.8 The Existence of “HTTPS” Token in the Domain Part of the URL

The phishers may add the “HTTPS” token to the domain part of a URL in order to trick users. For example, http://https-www-paypal-it-webapps-mpp-home.soft-hair.com/.

4.2.9 Website Traffic

This feature measures the popularity of the website by determining the number of visitors and the number of pages they visit. However, since phishing websites live for a short period of time, they may not be recognized by the Alexa database (Alexa the Web Information Company., 1996). By reviewing our dataset, we find that in worst scenarios, legitimate websites ranked among the top 100,000. Furthermore, if the domain has no traffic or is not recognized by the Alexa database, it is classified as “Phishing”. Otherwise, it is classified as “Suspicious”.

4.2.10 Domain Registration Length

Based on the fact that a phishing website lives for a short period of time, we believe that trustworthy domains are regularly paid for several years in advance. In our dataset, we find that the longest fraudulent domains have been used for one year only.

4.3 DECISION TREE

After feature extraction, we imported both the datasets, removed unnecessary columns like protocol and path. Both the datasets were merged into a single dataset and shuffled. Followed by this, the labels column was exported to store separately for calculating accuracy of the model. For decision tree algorithm, the dataset was split in 80:20 ratio for training and testing. The training dataset contained 1612 URLs and the testing dataset contained 403 URLs. Out of the URLs used for training, 810 were legitimate and 802 were phishing. In testing dataset, 207 were legitimate and 196 were phishing sites.

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. The accuracy received using this algorithm was 82.87%.

4.4 RANDOM FOREST

Random forest also went through a similar pre-processing like decision tree. After feature extraction, we imported both the datasets, removed unnecessary columns like protocol and path. Both the datasets were merged into a single dataset and shuffled. Followed by this, the labels column was exported to store separately for calculating accuracy of the model. For decision tree algorithm, the dataset was split in 70:30 ratio for training and testing. The training dataset contained 1410 URLs and the testing dataset contained 605 URLs. Out of the URLs used for training, 707 were legitimate and 703 were phishing. In testing dataset, 310 were legitimate and 295 were phishing sites.

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. The accuracy received using the default settings of this algorithm was 83.8%. We increased the accuracy by a tad by specifying maximum depth and number of trees. The final accuracy of this algorithm was 84.46%.

4.5 SOURCE CODE SCREENSHOTS

Features Extraction

1.Long URL to Hide the Suspicious Part

If the length of the URL is greater than or equal 54 characters then the URL classified as phishing

0 --- indicates legitimate

1 --- indicates Phishing

2 --- indicates Suspicious

In [12]:

```
def long_url(l):
    if len(l) < 54:
        return 0
    elif len(l) >= 54 and len(l) <= 75:
        return 2
    return 1
```

In [13]:

```
#Add above result to dataset
splitted_data['long_url'] = raw_data['websites'].apply(long_url)
```

Feature-2

2.URL's having "@" Symbol

Using "@" symbol in the URL leads the browser to ignore everything preceding the "@" symbol and the real address often follows the "@" symbol.

0 --- indicates legitimate

1 --- indicates Phishing

In [15]:

```
def have_at_symbol(l):
    if "@" in l:
        return 1
    return 0
```

In [16]:

```
splitted_data['having_@_symbol'] = raw_data['websites'].apply(have_at_symbol)
```

3.Redirecting using "//"

0 --- indicates legitimate

1 --- indicates Phishing

In [18]:

```
def redirection(l):
    if "//" in l:
        return 1
    return 0
```

In [19]:

```
splitted_data['redirection_//_symbol'] = seperation_of_protocol[1].apply(redirection)
```

Feature-4

4.Adding Prefix or Suffix Separated by (-) to the Domain

1 --> indicates phishing

0 --> indicates legitimate

In [21]:

```
def prefix_suffix_seperation(l):
    if '-' in l:
        return 1
    return 0
```

In [22]:

```
splitted_data['prefix_suffix_seperation'] = seperation_domain_name['domain_name'].apply(pre
```

Feature - 5

5. Sub-Domain and Multi Sub-Domains

0 --- indicates legitimate

1 --- indicates Phishing

2 --- indicates Suspicious

In [24]:

```
def sub_domains(l):
    if l.count('.') < 3:
        return 0
    elif l.count('.') == 3:
        return 2
    return 1
```

In [25]:

```
splitted_data['sub_domains'] = splitted_data['domain_name'].apply(sub_domains)
```

Feature-6

6.Using the IP Address

1 --> indicates phishing

0 --> indicates legitimate

In [27]:

```
import re
def having_ip_address(url):
    match=re.search('(([01]?\\d\\d?|2[0-4]\\d|25[0-5])\\.([01]?\\d\\d?|2[0-4]\\d|25[0-5])\\
                    '((0x[0-9a-fA-F]{1,2})\\.(0x[0-9a-fA-F]{1,2})\\.(0x[0-9a-fA-F]{1,2})\\.
                    '(?:[a-fA-F0-9]{1,4}:){7}[a-fA-F0-9]{1,4}',url)     #Ipv6
    if match:
        #print match.group()
        return 1
    else:
        #print 'No matching pattern found'
        return 0
```

Feature-7

7.Using URL Shortening Services

1 --> indicates phishing

0 --> indicates legitimate

In [30]:

```
def shortening_service(url):
    match=re.search('bit\.ly|goo\.gl|shorte\.st|go2l\.ink|x\.co|ow\.ly|t\.co|tinyurl|tr\.im
                    'yfrog\.com|migre\.me|ff\.im|tiny\.cc|url4\.eu|twit\.ac|su\.pr|twurl\.n
                    'short\.to|BudURL\.com|ping\.fm|post\.ly|Just\.as|bkite\.com|snipr\.com
                    'doiop\.com|short\.ie|kl\.am|wp\.me|rubyurl\.com|om\.ly|to\.ly|bit\.do|
                    'db\.tt|qr\.ae|adf\.ly|goo\.gl|bitly\.com|cur\.lv|tinyurl\.com|ow\.ly|b
                    'q\.gs|is\.gd|po\.st|bc\.vc|twitthis\.com|u\.to|j\.mp|buzurl\.com|cutt\
                    'x\.co|prettylinkpro\.com|scrnch\.me|filoops\.info|vzturl\.com|qr\.net|
    if match:
        return 1
    else:
        return 0
```

In [31]:

```
splitted_data['shortening_service'] = raw_data['websites'].apply(shortening_service)
```

Feature - 8

8.The Existence of “HTTPS” Token in the Domain Part of the URL

In [33]:

```
def https_token(url):
    match=re.search('https://|http://',url)
    if match.start(0)==0:
        url=url[match.end(0):]
    match=re.search('http|https',url)
    if match:
        return 1
    else:
        return 0
```

In [34]:

```
splitted_data['https_token'] = raw_data['websites'].apply(https_token)
```

Feature - 9

9.Website Traffic

Phishing sites may not be recognized by the Alexa database (Alexa the Web Information Company).

IF{Website Rank<100,000 → LegitimateWebsite Rank>100,

In [36]:

```
from bs4 import BeautifulSoup
import urllib.request
def web_traffic(url):
    try:
        rank = BeautifulSoup(urllib.request.urlopen("http://data.alexa.com/data?cli=10&dat=
    except TypeError:
        return 1
    rank= int(rank)
    if (rank<100000):
        return 0
    else:
        return 2
```

In [37]:

```
splitted_data['web_traffic'] = raw_data['websites'].apply(web_traffic)
```

Feature - 10

10.Domain Registration Length

IF{Domains Expires on≤ 1 years → Phishing

Otherwise→ Legitimate

In [47]:

```
import whois
from datetime import datetime
import time
def domain_registration_length_sub(domain):
    expiration_date = domain.expiration_date
    today = time.strftime('%Y-%m-%d')
    today = datetime.strptime(today, '%Y-%m-%d')
    if expiration_date is None:
        return 1
    elif type(expiration_date) is list or type(today) is list :
        return 2
    else:
        registration_length = abs((expiration_date - today).days)
        if registration_length / 365 <= 1:
            return 1
        else:
            return 0
```

In [51]:

```
def domain_registration_length_main(domain):
    dns = 0
    try:
        domain_name = whois.whois(domain)
    except:
        dns = 1

    if dns == 1:
        return 1
    else:
        return domain_registration_length_sub(domain_name)
```

In [52]:

```
splitted_data['domain_registration_length'] = splitted_data['domain_name'].apply(domain_reg
```

Dataset head after feature extraction:

```
In [53]:
splitted_data.head()
```

Out[53]:

	protocol	domain_name	address	long_url	havi
0	http	www.emuck.com:3000	archive/egan.html	0	
1	http	danoday.com	summit.shtml	0	
2	http	groups.yahoo.com	group/voice_actor_appreciation/links/events_an...	1	
3	http	voice-international.com		0	
4	http	www.livinglegendsltd.com		0	

4.6 RESULTS

We used two algorithms for training our machine learning model – Decision Tree Algorithm and Random Forest Algorithm. The accuracy for decision tree algorithm was 82.87 and the accuracy of Random Forest reached a maximum of 84.46. The feature importance graph showed that about 50% of the websites were classified phishing based on only three features.

1. The length of the URL
2. Web Traffic ranking at the Alexa Database
3. Multiple subdomains in Domain Name

The features importance plot of our model is as shown below:

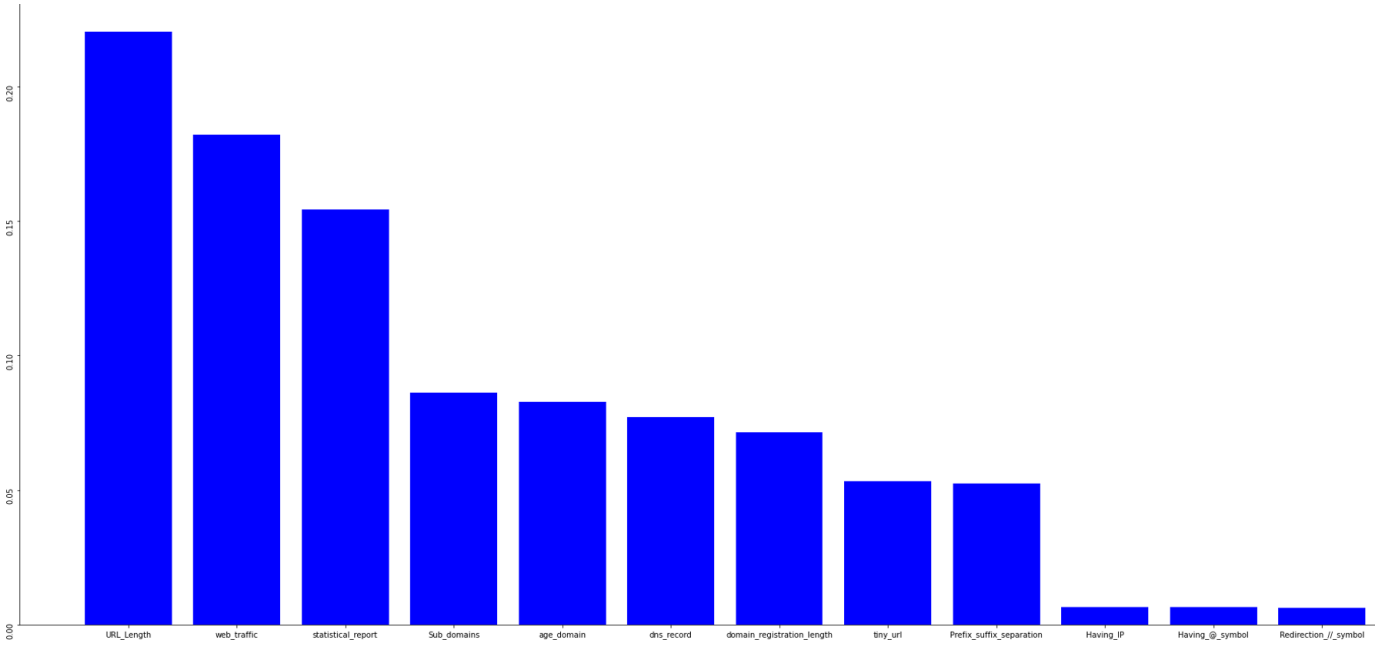

Fig 1: Feature Importance Plot

5. CONCLUSION

Cybercriminals and organizations with malicious intent are known to use phishing URLs. These URLs might collect information like usernames, passwords and various financial and banking details. Hackers might create such phishing sites to harvest personal data. The website is tried to fool users who are not technically sound. Phishing is known to even cost organizations a lot of money annually though the success rate on paper being low. Our project helps to distinguish between phishing, suspicious and legitimate URL's using a massive dataset provided by Alexa (to train). For phishing detection of URLs, we used two different Machine Learning models called Random Forest and Decision Tree which had different success rates as shown in the report. The accuracy in the Random Forest method was higher at 84.46 percent. Future work to this project may include – adding more features in feature extraction module to keep up with the various new and upcoming phishing techniques as cybercriminals evolve their techniques to exploit victims. Future work includes adding more features in feature extraction, checking page-based features after URL based features.

6. REFERENCES

Afroz, S. and Greenstadt, R., 2011, September. Phishzoo: Detecting phishing websites by looking at them. In 2011 IEEE fifth international conference on semantic computing (pp. 368-375). IEEE.

Chatterjee, M. and Namin, A.S., 2019, July. Detecting phishing websites through deep reinforcement learning. In 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC) (Vol. 2, pp. 227-232). IEEE.

Babagoli, M., Aghababa, M.P. and Solouk, V., 2019. Heuristic nonlinear regression strategy for detecting phishing websites. Soft Computing, 23(12), pp.4315-4327.

Ramesh, G., Krishnamurthi, I. and Kumar, K.S.S., 2014. An efficacious method for detecting phishing webpages through target domain identification. Decision Support Systems, 61, pp.12-22.

Srinivasa Rao, R. and Pais, A.R., 2017, April. Detecting phishing websites using automation of human behavior. In Proceedings of the 3rd ACM Workshop on Cyber-Physical System Security (pp. 33-42).

Jain, A. K., & Gupta, B. B. (2017). Phishing Detection: Analysis of Visual Similarity Based Approaches. Security and Communication Networks, 2017, 1–20. doi:10.1155/2017/5421046

Butnaru, A.; Mylonas, A.; Pitropakis, N. Towards Lightweight URL-Based Phishing Detection. Future Internet 2021, 13, 154.

Khonji, M., Iraqi, Y., & Jones, A. (2013). Phishing Detection: A Literature Survey. IEEE Communications Surveys & Tutorials, 15(4), 2091–2121.

Fette, I., Sadeh, N. and Tomasic, A., 2007, May. Learning to detect phishing emails. In Proceedings of the 16th international conference on World Wide Web (pp. 649-656).

Corona, I., Biggio, B., Contini, M., Piras, L., Corda, R., Mereu, M., Mureddu, G., Ariu, D. and Roli, F., 2017, September. Deltaphish: Detecting phishing webpages in compromised websites. In European Symposium on Research in Computer Security (pp. 370-388). Springer, Cham

Joby James, Sandhya l, Ciza Thomas,“Detection of Phishing URLs Using Machine Learning Techniques” Publication Year: 2013

Vaibhav Patil, Pritesh Thakkar, Chirag Shah, Tushar Bhat, Prof. S.P. Godse,”Detection and Prevention of phishing websites using machine learning Approaches” Publicate date: August, 2018

Neda Abdelhamid, Fadi Thabtah and Hussein Abdel-jaber,” Phishing Detection: A Recent Intelligent Machine Learning Comparison based on Models Content and Features” year of publication: july, 2017

Amani Alswailem, Bashayr Alabdullah, Norah Alrumayh and Dr.Aram Alsedrani, ” Detecting Phishing Websites Using Machine Learning” Year of publication: May, 2019.

Mohammed Hazim Alkawaz, Stephanie Joanne Steven and Asif Iqbal Hajamydeen,” Detecting Phishing Website Using Machine Learning” Year of publication: February,2020