Project Report

1. **INTRODUTION**
   - **Project Overview**
   - **Purpose**

2. **LITERATURE SURVEY**
   - **Existing problem**
   - **References**
   - **Problem Statement Definition**

3. **IDEATION & PROPOSED SOLUTION**
   - **Empathy Map Canvas**
   - **Ideation & Brainstorming**
   - **Proposed Solution**
   - **Problem Solution Fit**

4. **REQUIREMENT ANALYSIS**
   - **Functional Requirement**
   - **Non-Functional Requirements**

5. **PROJECT DESIGN**
   - **Data Flow Diagrams**
   - **Solution & Technical Architecture**

6. **PROJECT PLANNING & SCHEDULING**
   - **Sprint Planning & Estimation**
   - **Sprint Delivery Schedule**
   - **Reports From JIRA**

7. **CODING & SOLUTIONING**
   - **Feature 1**
   - **Feature 2**
   - **Database Schema (if applicable)**

1. INTRODUCTION

a. Project Overview

Phishing can be defined as impersonating a valid site to trick users by stealingtheir personal data comprising usernames, passwords, accounts numbers,national insurance numbers,etc. Phishing frauds might be the most widespread cybercrime used today. There are countless domains where phishing attack can occur like the online paymentsector, webmail, financial institutions, file hostingor cloud storageand many others. The webmail and online payment sector was embattled by phishing more than in any other industry sector. Phishing can be done through email phishing scams and spear phishing hence user should be aware of the consequences and should not give their

100 percent trust on common security application. MachineLearning is one of the efficienttechniques to detectphishingas it removes drawback of existing approach.

b. Purpose

The objectives which is the most vital thing in proposed project is to verify the validity of the website by capturing blacklisted URLs. To notify the user on blacklisted website through pop-upwhile they are trying to access and to notifythe user on blacklisted website through email while they are trying to access. This proposedproject will allowadministrator to add blacklisted URL's inorder to alertuser during their inquiry.

The two scope of project, which is well known as user scope and system scope. User has some responsibility towardsthe system. The system includesa few standards and policies that requires to be obliged in order to comply the system. The user can be notified if blacklisted website is being accessed. The admin can capture the blacklisted URL's to alert user. The system involves features like capturing blacklisted website, viewing blacklisted website, displaying pop-up notification and also displaying email notification.

2. LITERATURE SURVEY

a. Existing problem

Couple of researchers have analysed the stats of malicious sites in some way. Our method picks up some of the important ideas from previouscase studies. Ma, et al. [3,4] comparedvarious batch-based learningalgorithms used in classifying phishingsites and statedthat a combination of host based and lexicalbased featuresoutcome in the highest accuracy in classification. Besides, they are also compared with the performance of batch-based algorithms with the onlinebased algorithms which when utilizes complete features and noticed that onlinebased

algorithms, especially Confidence-Weighted (CW), stand out performing batch-based algorithms. The attributes include the existence of the red flag keywordspresent in the website, attributes that are based on Google's Page Rank and Google's Web page quality guidelines. One cannot compare directly without access to the same websites and attributes.

b. References

1. S. Sheng, M. Holbrook, P. Kumaraguru, L. F. Cranor, and J. Downs, "Who falls for phish?: a demographic analysis of phishing susceptibility and effectiveness of interventions," in Proceedings of the 28th international conference on Human factorsin computing systems,ser. CHI '10. New York,NY,USA: ACM, 2010, pp. 373–382.
2. B. Krebs, "HBGary Federalhacked by Anonymous," http: //krebsonsecurity.com/2011/02/hbgary-federal-hacked-by-anonymous/, 2011, accessed December2011.
3. B. Schneier, "Lockheed Martin hack linked to RSA's SecurID breach," http://www.schneier.com/blog/archives/2011/05/lockheed_martin.html, 2011, accessedDecember 2011.
4. C. Whittaker, B. Ryner,and M. Nazif, "Large-scale automatic classification of phishingpages," in NDSS '10, 2010.
5. X. Dong, J. Clark,and J. Jacob, "Modelling user-phishing interaction," in Human SystemInteractions, 2008 Conference on, may 2008, pp. 627 –632.
6. W. D. Yu, S. Nargundkar, and N. Tiruthani, "A phishing vulnerability analysis of web basedsystems," in Proceedings of the 13th IEEE Symposium on Computers and Communications (ISCC 2008). Marrakech, Morocco: IEEE, July 2008, pp. 326–331.
7. Anti-Phishing Working Group (APWG), "Phishing activity trends report — second half2010," http://apwg.org/reports/apwg_report h2 2010. pdf, 2010, accessedDecember 2011.
8. Anti-Phishing Working Group (APWG), "Phishing activity trends report — first half 2011,"http://apwg.org/reports/apwg_trends report h1 2011.pdf, 2011, accessedDecember 2011.
9. Anti-Phishing Working Group (APWG), "Phishing activity trends report

— second half2011," http://apwg.org/reports/apwg_trends report h2 2011.pdf, 2011, accessed July 2012.
10. B. Schneier, "Detailsof the RSA hack," http://www.schneier.com/blog/ archives/2011/08/details of the.html, 2011, accessed December2011

C. Problem Statement Definition

Phishing detection techniques do suffer low detection accuracy and high false alarm especially when novel phishing approaches are introduced. Besides, the most common technique used, blacklist-based method is inefficient in responding to emanating phishing attacks since registering

new domain has become easier, no comprehensive blacklist can ensure a perfect up-to-date database. Furthermore, page content inspection has been used by some strategies to overcome the false negative problems and complement the vulnerabilities of the stale lists.

Moreover, page content inspection algorithms each have different approach to phishing website detection with varying degrees of accuracy.Therefore, ensemble can be seen to be a better solution

as it can combine the similarity in accuracy and different error-detection rate properties in selectedalgorithms. Therefore, this study will address a couple of research:

1. How to process raw dataset for phishing detection?
2. How to increase detection rate in phishing websites algorithms?
3. How to reducefalse negative rate in phishingwebsites algorithm?
4. What are the best compositions ofclassifiers that can give a good detection rate ofphishing website?

1. IDEATION & PROPOSEDSOLUTION

a. Empathy Map Canvas

b. Ideation &Brainstorming

c. Proposed Solution

S. No.	Parameter	Description
1.	**Problem Statement(Problem to be solved)**	Phishing sites are malicious website that aim to steal user's personal data. Spotting these phishing website is typically a challenging task because phishing is mainly a semantic-based attack that mainly focusedon softwarevulnerability etc.

2.	**Idea/Solutiondescription**	Our product server as a browser extension and it scrapesthe website URL and runs it through our ML model. If the modeldetects it as a phishing website, the extension notifies the user.

3.	**Novelty / Uniqueness**	The browser extension factors is not used in any previous works. Theuser does not have to think twicebefore using a website,our extention will take careof the classifying work.
4.	**SocialImpact /CustomerSatisfaction**	Reduce the amount of information stolenby phishing sitesand also increase customer satisfaction as they would be reassured when using legitimate website.
5.	**BusinessModel(Revenue Model)**	We propose a two tier system namely a FREE and PREMIUM tire. The FREE tier would include ads and the PREMIUM tier is a recurring subscription either monthly or annually.

6.	**Scalability of theSolution**	Since this is a browser extension which would be published in Chrome Marketplace,it can be accessed andused by everyone across the world.

d. **Problem Solutionfit**

2. REQUIREMENT ANALYSIS

a. Functional requirement

FR No.	Functional Requirement(Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Learning &Detection	The samples and the topological structure of themachine learning TensorFlow is built. The submittedURLsare tested against thesamples in the database toperform classification.
FR-2	Testing & Alert	URLs passed throughthe system are recorded in a database, thus each URL submitted by the user is tested to check or duplicate. If a phishing website is detected the popup message will alertthe user. Give information aboutthe malicious websitewith accurate result.
FR-3	Deep Learning	The phishing detection process couldbe doneusing theRecurrent Neural Network.The website could bedetected.
FR-4	Hardware Requirements	2GB RAM(minimum) 100GB HDD(minimum)Inteli3 quad core 1.66GHz processor(minimum) InternetConnectivity
FR-5	Software Requirements	Windows 7 or higher Python 3.6.0 or higher Visual StudioCode Flask(python platform) HTML Dataset consisting of Phishing websites and their features. Required plugins and libraries Jupiter notebook
FR-6	Other requirements	IBM cloud login Chrome extension features

b. Non-Functional requirements

FR No.	Non-Functional Requirement	Description
NF R-1	**Usability**	This system is really used as it can able to detectphishing websites. By detecting malicious websites, our personal andprofessional data are confidential, secure,and accessible.

NF R-2	**Security**	Phishers spoof legitimate emails so that the victim trusts them.They send out massive numbers of fraudulent emails in order to catch a smallpercentage of recipients off guard. They createa sense of urgency so that the victim doesnot think twicebefore clicking the link or downloading the attachment. Lack of security awareness among employeesis alsoone of the major reasonsfor the successof phishing. Organizations should be awareofhow the benefits and purpose of security awareness training can secure their employees from fallingvictim to phishingattacks.
NF R-3	**Reliability**	The performance of the systemwouldbe accurate. Probability of giving falseinformation is verylow. As the systemis working based on the deep learning algorithm, it would easily predict and give the perfect information.

NF R-4	**Performance**	The effectiveness of these methods relies on featurecollection, training data,and classification algorithms and giving alerts whenphished websites are detected. It must be processed and executed within a fraction of a second using the deep learning algorithm
NF R-5	**Availability**	The availability of the solution is effective and it should be helpful in a great way to prevent our personal data to be exposed.

NF R-6	**Scalability**	This solution is scalable enoughto fit theSecurity issues by constructing the best website. The cost of establishing the website and maintaining all the programs may be high . It is acceptable to fit them over any place and any resources.

3. PROJECT DESIGN

a. Data Flow Diagrams

b. **Solution &Technical Architecture**

4. PROJECT PLANNING & SCHEDULING

a. Sprint Planning & Estimation

Sprint	Functional Requirement(Epic)	User Story Number	User Story/ Task	StoryPoints	Priority	Team Members
Sprint-1	URL detector	USN-1	URL is the first thing to analyze a websiteto decide whetherit is a phishing or not	10	High	M.Dhinesh R.Jefril Angelan K.S.Gokula Krishna R.Gunasekaran

Sprint-1		USN-2	Some of URL-Based Features are 1. Digit count in theURL 2. Total length of URL 3. Checking whetherthe URL is typo- squatted or not 4. Checking whether itincludes a legitimate brand nameor not 5. Number of subdomains in URL 6. TLD is one of the commonly used one	10	High	M.Dhinesh R.Jefril Angelan K.S.Gokula Krishna R.Gunasekaran
Sprint-2	Domain detection	USN-3	The purpose of Phishing DomainDetection is detecting phishing domain names.Therefore, passive queries related to the domain name,which we want to classify as phishing or not, provide useful information tous.	10	High	M.Dhinesh R.Jefril Angelan K.S.Gokula Krishna R.Gunasekaran

Sprint-2		USN-4	Some useful Domain-Based Features are 1. Its domain name or its IP address in blacklists ofwellknown reputation services? 1. How many days passed since the doma	10	High	M.Dhinesh R.Jefril Angelan K.S.GokulaKrishna R.Gunasekaran

			in was registered? 1. Is the registrant name hidden?			

Sprint-3	Page basedfeatures and C o n t e n t ba s e d features	USN-5	Page-Based Features are using information aboutpages w h i c h a r e c a l c u la t e d reputation ranking services. Obtaining these typesof features requires active scan to target domain. Page contents are processed for us to detectwheth er target domain is used for phishing or not	10	High	M.Dhinesh R.Jefril Angelan K.S.Gok ulaKrishna R.Gunasekar an

Sprint-3						M.Dhinesh R.Jefril Angelan K.S.Gok ula Krishna R.Gunasekar an

			1. Global pagerank 2. Country pagerank 1. Position at the Alexa top 1 million siteSome processed information about pages are 1. Page titles 1. Meta tags 1. Hidden text ● Text inthe body ● Images etc.			

Sprint-4	D e t e c t i on process	USN-6	Detecting Phishing Domains is a classificati on problem, so it means we need labeled data which has samples as phish domains and legitimate doma i n s i n t h e t r a i n i n g p h a s e	20		M.Dhinesh R.Jefril Angelan K.S.Gok ula Krishna R.Gunasekar an

Project Tracker, Velocity &Burndown Chart:

Sprint	Total StoryPoints	Duration	SprintStart Date	Sprint End Date(Planned)	Story Points Completed (as onPlanned End Date)	Sprint ReleaseDate (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	10	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	10	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	10	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

Velocity:

Imagine we have a 10-day sprint duration,and the velocity of the team is 20 (points per sprint). Let's calculatetheteam's averagevelocity (AV) per iteration unit (story pointsper day)

Burndown Chart:

A burndown chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurableprogress over time.

b. Sprint Delivery Schedule

Sprint	Sprint Topic	S t a r t Date	Expected Delivery
Sprint 1	URL detector	24-10-2022	29-10-2022
Sprint 2	Domain detection	31-10-2022	05-11-2022
Sprint 3	Page based features and content basedfeatures	07-11-2022	12-11-2022
Sprint 4	Detection process	14-11-2022	19-11-2022

c. Reports from JIRA

5. CODING & SOLUTIONING

a. Feature 1

This feature is used to importrequired libraries to load the model from the .pkl file which was builded in the model building phase.

Coding :

```
from flask import Flask, request,
render_templateimport numpy as
np
import
pandas as
pd from
sklearn
importme
trics
```

```
import
warnings
import
pickle
warnings.filterw
arnings('ignore')
from feature
import
FeatureExtracti
on

file =
open("model
.pkl","rb")
gbc =
pickle.load(f
ile)
file.close()



app= Flask( name )

@app.route("/",
methods=["GET",
"POST"]) def index():
if request.method
== "POST":

      url =
request.form[
"url"]obj =
FeatureExtra
ction(url)
```

```
        x = np.array(obj.getFeaturesList()).reshape(1,30)


        y_pred
        =gbc.pre
        dict(x)[0]
        #1 is safe
    #-1 is unsafe
        y_pro_phishing              =
    gbc.predict_proba(x)[0,0]
    y_pro_non_phishing              =
    gbc.predict_proba(x)[0,1]       #
    if(y_pred ==1 ):
        pred = "Itis {0:.2f} % safe to go ".format(y_pro_phishing*100)
                                                return
                                                render_template('index.html',xx
    =round(y_pro_non_phishing
    ,2),url=url         )       return
    render_template("index.html
    ", xx =-1)



    if__name__== "_
       main ":
       app.run(debu
       g=True,port=
       2002)
```

b. Feature 2

This feature helps in providing easy UI to the user using the web interface.Coding:

```
<!DOCTYPE html>
```

```
<html lang="en">

<head>
   <center> <h1> IBM Project Based Learning</h1> </center>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta name="description" content="This website is developfor
identify thesafety of url.">
   <meta name="keywords" content="phishing
url,phishing,cybersecurity,machine
learning,classifier,python">
   <meta name="author" content="Balajee A V">


   <!-- BootStrap -->
   <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/
4.5.0/css/bootstrap.min.css"
      integrity="sha384-
9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MY
YxFfc+NcPb1d KGj7Sk" crossorigin="anonymous">

   <link href="static/styles.css" rel="stylesheet">
   <title>URL detection</title>      </head>


<body>
 <center> <img class="image image-contain"
src="https://cdn.activestate.com/wp-
content/uploads/2021/02/phishing-detection-with-Python.jpg"
alt="MDN logo" /> </center>
```

```
<div class=" container">
   <div class="row">
      <div class="form col-md" id="form1">
         <h2>PHISHING URL DETECTION</h2>


         <br>
         <form action="/" method ="post">
            <input type="text" class="form input"
name ='url' id="url"placeholder="Enter URL"
required="" />

            <label for="url" class="form_label">URL</label>
            <button class="button" role="button" >Checkhere</button>
</form>


   </div>


   <div class="col-md" id="form2">


      <br>
      <h6 class = "right "><a href= {{ url }} target="_blank">{{ url
      }}</a></h6>


      <br>
      <h3 id="prediction"></h3>
      <button class="button2" id="button2" role="button"
onclick="window.open('{{url}}')" target="_blank" >Still
want to Continue</button>                              <button
class="button1" id="button1" role="button"
onclick="window.open('{{url}}')"
```

```
target="_blank">Continue</button>
    </div>
</div>
<br>
</div>


    <!-- JavaScript -->
    <script     src="https://code.jquery.com/jquery-
3.5.1.slim.min.js"                integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamo-
FVy38MVBnE+IbbVYUew+OrCXaRkfj"
        crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/um
d/popper.min.js" integrity="sha384-
Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtm
I3UksdQRVvoxMf ooAo"
        crossorigin="anonymous"></script>
    <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/b
ootstrap.min.js"

        integrity="sha384-
OgVRvuATP1z7JjHLkuOU7Xw704+h835Lr+6QL9UvYjZE3
Ipu6Tp75j7Bh/kR 0JKI"
        crossorigin="anonymous"></script>



    <script>

            let x = '{{xx}}';
```

```
let num =
x*100;
if(0<=x
&&x<0.5
0){
            num = 100-num;
        }
        let txtx =
num.toString();
if(x<=1 &&
x>=0.50){
            var label = "Website is "+txtx +"%
safe to use...";
document.getElementById("prediction").inne
rHTML = label;
document.getElementById("button1").style.d
isplay="block";
        }
        else if (0<=x && x<0.50){
            var label = "Website is "+txtx
+"%unsafe to use..."
document.getElementById("prediction").inne
rHTML = label ;
document.getElementById("button2").style.di
splay="block";
        }

   </script>

</body>
</html>
```

6. TESTING

a. Test Cases

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	Test for Automation (Y/N)	BUG ID	Executed By
LoginPage_ TC_OO1	UI	Home Page	Verify the UIelements is Responsive		1. Enter URL and click go	https://www.google.com/	Should Wait for Response and then gets Acknowledge	Working as expected	Pass		N		M.Dhinesh

					2. Type or copy paste the URL 3. Check whether the button is responsive or not 4. Reload and Test Simultaneously								

LoginPage_TC_OO2	Functional	Home page	Verify whether the link is legitimate or not		1. Enter URL and click go 2. Type or copy paste the URL 3. Check the website is legitimate or not 4. Observe the Results	https://www.youtube.com/	User should observe whether the website is legitimate or not.	Working as expected	Pass		N		R.Jefril Angelan

LoginPage_TC_OO3	Functional	Home Page	Verify user is able to access the legitimate website or not		1. Enter URL and click go 2. Type or copy paste the URL 3. Check the website is legitimate or not 4. Continue if the website is legitimate or be cautious if it is not legitimate.	http://ssalescript.info/	Application should show that Safe Webpage or Unsafe.	Working as expected	Pass		N		K.S.Gokul Krishna

LoginPage_TC_OO4	Functional	Home Page	Testing thewebsitewith multipleURLs		1. Enter URL (https://phishing-shield.herokuapp.com/) and click go 2. Type or copy paste the URL totest 3. Check the website	https://www.delgets.com/	User can able to identify the website whether it is secure or not	Working as expected	Pass		N		R.Gunasekaran

					is legiti mate or not 4 . Contin u e i f t h websit e is secure or be cautious if it is no t secure								

b. User Acceptance Testing

1. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level,and how they wereresolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	10	2	4	20	36
Not Reproduced	0	0	1	0	1
Skipped	0	0	0	0	0

Won't Fix	0	0	2	1	3
Totals	23	9	12	25	60

1. **Test Case Analysis**

This reportshows the numberof test cases that have passed,failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	10	0	0	10
Client Application	50	0	0	50
Security	5	0	0	4
Outsource Shipping	3	0	0	3
Exception Reporting	10	0	0	9
Final Report Output	10	0	0	10
Version Control	4	0	0	4

7. RESULTS

a. **Performance Metrics**

8. ADVANTAGES & DISADVANTAGES

Advantages:

1. This system can be used by many E-commerceor other websites in order to have

   good customer relationship.

2. User can make onlinepayment securely.
3. Data mining algorithm used in this system provides better performance as comparedto other traditional classifications algorithms.
4. With the help of this system user can also purchase productsonline without anyhesitation.

Disadvantages

5. If Internet connection fails, this system won't work.
6. All websites related data will be stored in one place.

9. CONCLUSION

It is outstanding that a decent enemy of phishingapparatus ought to anticipate the phishing assaults in a decent timescale. We accept that the accessibility of a decentenemy of phishingdevice at a decent time scale is additionally imperative to build the extent of anticipating phishing sites. This apparatus ought to be improved continually through consistent retraining. As a matter of fact,the accessibility of crisp and cutting-edge preparing dataset which may gained utilizing our very own device [30, 32] will help us to retrain our model consistently and handle any adjustments in the highlights, which are influential in deciding the site class. Albeit neural system demonstrates its capacity to tackle a wide assortment of classification issues, the procedure of

finding the ideal structure is very difficult, and much of the time, this structure is controlled by experimentation.Our model takes care of this issue via computerizing the way toward organizing a neural system conspire; hence, on the off chance that we construct an enemy of phishing model and for any reasons we have to refresh it, at that point our model will encourage this procedure, that is, since our model will mechanize the organizing procedure and will request scarcely any client defined parameters.

10. FUTURE SCOPE

In future if we get structured dataset of phishing we can perform phishing detection much more faster than any other technique.In future we can use a combination of any other two or more classifier to get maximumaccuracy. We also plan to explorevarious phishing techniques that uses Lexical features, Network based features,Content based features, Webpage based features and HTML and JavaScript features of web pages which can improvethe performance of the system. In particular, we extractfeatures from URLs and pass it through the various classifiers.

11. APPENDIX

A mechanism to detect phishingwebsites. Our methodology uses not just traditional URL based or content based rules but rather employs the machine learning technique to identify not so obvious patterns and relations in the data. We have used features from various domain spanning from URL to HTML tags of the webpage, from embedded URLs to favicon,and databases like WHOIS, Alexa, Pagerank, etc. to check the traffic and status of the website. We were able to obtain an

accuracy of more than 96%, recall greater than 96% with a False Positive Rate of less than 5%, thus classifying most websites correctly and proving the effectiveness of the machine learning based technique to attack the problem of phishing websites. We provided the output as a user-friendly web platform which can further be extended to a browser extension to provide safe and healthyonline space to the users.

Source Code:

```
import ipaddress
import re import
urllib.request
from bs4 import
BeautifulSoup
import socket
import requests
from
googlesearch
importsearch
import whois
from datetime import date,
datetime import time from
dateutil.parser import parse
as date_parse from
urllib.parse import urlparse


class FeatureExtraction:
features = []    def
```

```
__init_(self,url):
self.features = []        self.url
= url
self.doma
in = ""
self.whois
_response
= ""
self.urlpar
se = ""
self.respo
nse = ""
self.soup=
""


        try:
            self.response = requests.get(url)
self.soup =BeautifulSoup(response.text,
'html.parser')
except: pass


        try:
            self.urlpar
se = urlparse(url)
self.domain =
```

```
self.urlparse.netl
oc except:
pass
        try:
            self.whois_response =
whois.whois(self.domain)except:
pass
self.features.append(sel
f.UsingIp())
self.features.append(sel
f.longUrl())
self.features.append(sel
f.shortUrl())
self.features.append(sel
f.symbol())
self.features.append(sel
f.redirecting())
self.features.append(sel
f.prefixSuffix())

self.features.append(self.SubDo
mains())
self.features.append(self.Hppts(
))
self.features.append(self.Domain
RegLen())
```

```
self.features.append(self.Favicon
())
self.features.append(self.NonStd
Port())
self.features.append(self.HTTPS
DomainURL())
self.features.append(self.Request
URL())
self.features.append(self.Anchor
URL())
self.features.append(self.LinksIn
ScriptTags())
self.features.append(self.ServerF
ormHandler())
self.features.append(self.InfoEm
ail())
self.features.append(self.Abnor
malURL())
self.features.append(self.Website
Forwarding())
self.features.append(self.StatusB
arCust())
self.features.append(self.Disable
RightClick())
self.features.append(self.UsingP
opupWindow())
```

```
self.features.append(self.Iframe
Redirection())
self.features.append(self.AgeofD
omain())
self.features.append(self.DNSRe
cording())
self.features.append(self.Website
Traffic())
self.features.append(self.PageRa
nk())
self.features.append(self.Google
Index())
self.features.append(self.LinksPo
intingToPage())
self.features.append(self.StatsRe
port())
    #
1
.
U
si
n
g
I
p
d
```

e

f

U

si

n

g

I

p

(

s

e

lf

):

tr

y

:

ipaddress.ip_

address(self.url)

return-1

except:

return 1

#

2.lo

ngU

```
rl
def
long
Url(
self
):  if
len(s
elf.u
rl)  <
54:
        return        1        if
len(self.url)   >=   54   and
len(self.url) <= 75:
        return 0
return -1


  #
3.
s
h
o
rt
U
rl
d
```

ef

s

h

o

rt

U

rl

(s

el

f

):

match =

re.searc

h('bit\.ly

|goo

\.gl|

sho

rte\.

st|g

o2l\

.i

nk|

x\.c

o|o

w\.l

y|t\.

```
    co|
tinyurl|tr\.im|is\.gd|cli\.gs|'
                    'yfrog\.com|migre\.me|ff\.im|tiny\.cc|url4\.eu|twit\.a
c|su\.pr| twurl\.nl|snipurl\.com|'
                    'short\.to|BudURL\.com|ping\.fm|post\.ly|Just\.as|bkite\.com|
snipr\.com|fic\.kr|loopt\.us|'
                    'doiop\.com|short\.ie|kl\.am|wp\.me|rubyurl\.com|o
m\.ly|to\.ly| bit\.do|t\.co|lnkd\.in|'
                    'db\.tt|qr\.ae|adf\.ly|goo\.gl|bitly\.com|cur\.lv|tinyurl\.com|ow\.ly|
bit\.ly|ity\.im|'

                    'q\.gs|is\.gd|po\.st|bc\.vc|twitthis\.com|u\.to|j\.mp|bu
zurl\.com| cutt\.us|u\.bb|yourls\.org|'
                  'x\.co|prettylinkpro\.com|scrnch\.me|filoops\.info|vzturl\.com|qr\.n
et| 1url\.com|tweez\.me|v\.gd|tr\.im|link\.zip\.net',
    self.url)       if match:
    return -1                 return1


      #
    4.Sym
    bol@
    def
    symb
    ol(self
    ):    if
```

```
re.fin
dall("
@",se
lf.url):
        return -1
return 1


   #
5.Redire
cting//
def
redirectin
g(self):
if
self.url.rf
ind('//')>
6: return
-1
return1


   #
6.pref
ixSuf
fix
def
```

```
prefix
Suffi
x(self
):try:
        match =
re.findall('\-', self.domain)
if match:
return -1
return
1       except:         return -1


   # 7.SubDomains
def
SubDomains(self):
dot_count=

len(re.findall("\.",
self.url))
if dot_count == 1:
return 1
elif
dot_count == 2:
        return 0
return -1
```

```
    # 8.HTTPS
d
ef
H
pp
ts(
sel
f):
tr
y:
            https =
self.urlparse.sche
meif 'https' in
https:
                return 1
ret
u
rn
-1
ex
ce
pt:
return 1


    #
```

9.Domai

nRegLen

def

DomainR

egLen(sel

f):try:

expiration_date =

self.whois_response.expiration_date

creation_date =

self.whois_response.creation_date

try:if(len(expiration_date)):

expiration_date = expiration_date[0]

except:

pass try:

if(len(creation_d

ate)):

creation_date = creation_date[0]

except: pass

age = (expiration_date.year-

creation_date.year)*12+ (expiration_-date.month-

creation_date.month)

if age >=12: return

1

re

tu

```
rn
-1
e
x
c
e
p
t:
        return -1


    # 10. Favicon    def Favicon(self):
try:                        for head in
self.soup.find_all('head'):
for head.link inself.soup.find_all('link',
href=True):
                dots = [x.start(0) for x in re.finditer('\.',
head.link['href'])]
ifself.url in head.link['href'] or len(dots) == 1 or domain in
head.link['href']:
                    return 1
return -1       except:
return -1


    #
11.
```

```
Non
StdP
ort
def
Non
StdP
ort(s
elf):
try:
        port=
self.domain.split("
:")if len(port)>1:
return -1 return 1
except:
return
-1


    # 12.
HTTPSDomai
   nURL
     def
HTTPSDomai
 nURL(self):

try:
if
'https'
```

```
in
self.d
omai
n:
                return -1
return 1        except:
            return -1


    # 13. RequestURL
def RequestURL(self):try:  for
img in self.soup.find_all('img',
src=True):
                dots = [x.start(0) for x in re.finditer('\.',
img['src'])]
ifself.url in img['src'] or self.domain in img['src'] or
len(dots) == 1:
                    success = success + 1
i = i+1


            for audio in self.soup.find_all('audio', src=True):
                dots = [x.start(0) for x in re.finditer('\.', audio['src'])]
if self.urlin audio['src'] or self.domain in audio['src'] or len(dots)
== 1:
                    success = success+ 1
i = i+1
```

```
            for embed in self.soup.find_all('embed', src=True):
                dots = [x.start(0) for x in re.finditer('\.',
embed['src'])]
ifself.url in embed['src'] or self.domain in embed['src'] or len(dots)
== 1:
                    success = success + 1
    i = i+1


            for iframe in self.soup.find_all('iframe', src=True):

                dots = [x.start(0) for x in re.finditer('\.',
iframe['src'])]
ifself.url in iframe['src'] or self.domain in iframe['src'] or len(dots)
== 1:
                    success = success + 1
    i = i+1


            try:
                percentage =
    success/float(i) * 100if
    percentage < 22.0:
                    return 1             elif((percentage >= 22.0) and
    (percentage < 61.0)):                return 0             else:
```

```
                return -1
    except:
    return 0        except:
            return -1


      # 14.
    Anchor
    URLdef
    Anchor
    URL(sel
    f):try:
            i,unsafe = 0,0
            for a in self.soup.find_all('a', href=True):
                if "#" in a['href'] or "javascript" in
a['href'].lower() or "mailto" ina['href'].lower() or not (url in
a['href'] or self.domain in a['href']):
                    unsafe = unsafe + 1
    i = i + 1


            try:
                percentage =
    unsafe / float(i) * 100if
    percentage < 31.0:
```

```
                return 1            elif
((percentage >= 31.0) and(percentage <
67.0)):                             return 0
else:
                return -1
except:
            return -1


      ex
c
e
p
t
:

r
e
t
u
r
n

-
1
```

```
    # 15.
LinksInScri
ptTags def
LinksInScri
ptTags(self
):try:
        i,success = 0,0

        for link in self.soup.find_all('link', href=True):
            dots = [x.start(0) for x in re.finditer('\.', link['href'])]
            if self.url in link['href'] or self.domain in link['href']
                or len(dots) == 1:success= success + 1
i = i+1

        for script in self.soup.find_all('script', src=True):
            dots = [x.start(0) for x in re.finditer('\.', script['src'])]
if self.urlin script['src'] or self.domain in script['src'] or len(dots)==
1:
                success = success + 1
i = i+1

        try:
```

```
                percentage =
success / float(i) * 100if
percentage < 17.0:
                    return 1              elif((percentage >= 17.0) and
(percentage < 81.0)):                 return 0              else:
                    return -1
except:
return 0        except:
return -1


   # 16.
ServerForm
Handler def
ServerForm
Handler(self
):try:
if
len(self.soup.
find_all('for
m',
action=True)
)==0:
return 1            else :
for form in
self.soup.find_a
```

```
        ll('form',
        action=True):
        if form['action']
        == "" or
        form['action']
        ==
        "about:blank":
        return-1


  tion']:


                                                                    else:

elif self.url not in form['action'] and self.domain not in form['ac-


    return 0


    return 1


        excep
        t:
        return
        -1

            # 17. InfoEmail    def
```

```
InfoEmail(self):       try:if
re.findall(r"[mail\(\)|mailto:?]",
self.soap):
                return -1

else:


except: return -1


return 1



   # 18.
Abnormal
URLdef
Abnormal
URL(self):
try: if
self.respon
se.text
==
self.whois
_response:
return 1
else:
```

```
                return -1
            except:
            return -1


              #19.
            WebsiteForwardi
            ng
            def
            WebsiteForwardin
            g(self):
            try:if
            len(self.response.h
            istory) <= 1:
            return 1
            elif
            len(self.response.history) <= 4:
                return 0

            else:


            except: return -1


return -1

              # 20. StatusBarCust    def StatusBarCust(self):
```

```
try:                    if
re.findall("<script>.+onmouseover.+</script>",
self.response.text): return 1                    else:
            ret
      u
      r
      n

      -
      1

      e
      x
      c
      e
      p
      t
      :

return -1


   # 21. DisableRightClick    def
DisableRightClick(self):try:  if
re.findall(r"event.button ?==
?2",self.response.text):
            return 1

else:


except: return -1
```

```
return -1


    # 22. UsingPopupWindow
    def
    UsingPopupWindow(self):
    try:                        if
    re.findall(r"alert\(",
    self.response.text):
            return 1

    else:


    except: return -1


return -1


    #23. IframeRedirection    def
    IframeRedirection(self):      try:if
    re.findall(r"[<iframe>|<frameBorder>]",
    self.response.text): return 1  else:
```

```
            return -1
ex
ce
pt:
ret
u
rn
-1


   # 24.
AgeofDo
maindef
AgeofDo
main(self)
:try:
        creation_date =
self.whois_response.creation_datetry:
if(len(creation_date)):
                creation_date = creation_date[0]
except:             pass


        today  = date.today()
age = (today.year-
creation_date.year)*12+(today.month-
creation_-
```

```
date.
    m
    o
    nt
    h)
    if
    a
    g
    e
    >
    =
    6:
    return 1
    ret
    u
    rn
    -1
    ex
    ce
    pt:
    return -1


      # 25.
    DNSRec
    ording
```

```
    def
    DNSRec
    ording(s
    elf):
    try:
            creation_date =
    self.whois_response.creation_datetry:
    if(len(creation_date)):
                    creation_date = creation_date[0]
    except:             pass

            today  = date.today()
    age = (today.year-
    creation_date.year)*12+(today.month-
    creation_-
date.
    m
    o
    nt
    h)
    if
    a
    g
    e
```

```
    >
    =
    6:
    return 1
    ret
    u
    rn
    -1
    ex
    ce
    pt:
    return -1


      #       26.
    WebsiteTr
    affic     def
    WebsiteTr
    affic(self):
    try:
            rank =
BeautifulSoup(urllib.request.urlopen("http://data.alexa.com/
data?cli=10&dat=s&url=" + url).read(),
"xml").find("REACH")['RANK']
    if (int(rank) < 100000):
            return 1
```

```
    ret
    u
    rn
    0
    ex
    ce
    pt
    :
    return -1
      #
    27.
    Pag
    eR
    ank
    def
    Pag
    eR
    ank
    (sel
    f):
    try:
            prank_checker_response =
requests.post("https://www.checkpager-ank.net/index.php",
{"name": self.domain})

            global_rank = int(re.findall(r"Global Rank: ([0-9]+)",
```

```
            rank_checker_re-sponse.text)[0])        if global_rank > 0 and
                global_rank < 100000:               return 1
                return -1       except:
                        return -1


                  #
                28.
                Googl
                eInd
                ex def
                Googl
                eInde
                x(self
                ):try:


                if site:


                else:


                except:
site = search(self.url, 5)


    return 1
```

```
return -1


            return 1


               # 29. LinksPointingToPage
               def
            LinksPointingTo
            Page(self):try:
                    number_of_links = len(re.findall(r"<a href=",
            self.response.text))if number_of_links == 0:
            return 1                                        elif
            number_of_links <= 2:              return 0
            else:
                       return -1
            except:
                    return -1

               #
            30.
            Stats
            Repo
            rt def
            Stats
            Repo
```

```
    rt(self
    ):try:
            url_match = re.search(
        'at\.ua|usa\.cc|baltazarpresentes\.com\.br|pe\.hu|esy\.es|hol\
        .es|
sweddy\.com|myjino\.ru|96\.lt|ow\.ly',  url)
ip_address                  =
socket.gethostbyname(self.domain)
ip_match
=re.search('146\.112\.61\.108|213\.174\.157\.151|
121\.50\.168\.88|192\.185\.217\.116|78\.46\.211\.158|181\.174\.165
\.13|
46\.242\.145\.103|121\.50\.168\.40|83\.125\.22\.219|46\.242\.145\.98|'
                    '107\.151\.148\.44|107\.151\.148\.107|64\.70\.19\.203|
199\.184\.144\.27|107\.151\.148\.108|107\.151\.148\.109|119\.28\.52\.61|
54\.83\.43\.69|52\.69\.166\.231|216\.58\.192\.225|'
                            '118\.184\.25\.86|67\.208\.74\.71|23\.253\.126\.
            58|
            104\.239\.157\.210|175\.126\.123\.219|141\.8\.224\.221|10\.10\.10\.
            10|
43\.229\.108\.32|103\.232\.215\.140|69\.172\.201\.153|'
'216\.218\.185\.162|54\.225\.104\.146|103\.243\.24\.98|
199\.59\.243\.120|31\.170\.160\.61|213\.19\.128\.77|62\.113\.226\.131|
208\.100\.26\.234|195\.16\.127\.102|195\.16\.127\.157|'
                            '34\.196\.13\.28|103\.224\.212\.222|172\.217\.4\.2
                25|
```

```
                54\.72\.9\.51|192\.64\.147\.141|198\.200\.56\.183|23\.253\.164\.1
                03|
52\.48\.191\.26|52\.214\.197\.72|87\.98\.255\.18|209\.99\.17\.27|'
                                    '216\.38\.62\.18|104\.130\.124\.96|47\.89\.58\.1
                41|
                78\.46\.211\.158|54\.86\.225\.156|54\.82\.156\.19|37\.157\.192\.1
                02|
204\.11\.56\.48|110\.34\.231\.42',
    ip_address)
    if url_match:
    return -1
    elif ip_match:

    return -1          return 1       except:
    return 1


       def
    getFeatures
    List(self):
    return
    self.featur
    es
```

GitHub link https://github.com/IBM-EPBL/IBM-Project-34894-1660279057