IBM - Project - 20179-1659714221

Project Documentation Report

Project Name : Web Phishing Detection

Team ID : PNT2022TMID27267

Team Members :

- Muazzam N Alseri [Team Lead]
- Mohamed Suhaib Ahmed
- Dilip Kumar K
- Kishore G

Github Link :
https://github.com/IBM-EPBL/IBM-Project-20179-1659714221#ibm-project-20179-1659714221

Demonstration Video Link :

https://youtu.be/f5lxCFdfPYI

1. Introduction:

1.1 Project Overview:

Phishing can be defined as impersonating a valid site to trick users by stealing their 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 attacks can occur like the online payment sector, webmail, financial institutions, file hosting or cloud storage and 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 users should be aware of the consequences and should not give their 100 percent trust on common security applications. Machine Learning is one of the efficient techniques to detect phishing as it removes the drawback of the existing approach. In addition to that, There are a number of users who purchase products online and make payments through e-banking. There are e-banking websites that ask users to provide sensitive data such as username, password & credit card details, etc., often for malicious reasons. This type of e-banking website is known as a phishing website. Web service is one of the key communications software services for the Internet. Web phishing is one of many security threats to web services on the Internet. Common threats of web phishing are :

- Web phishing aims to steal private information, such as usernames, passwords, and credit card details, by way of impersonating a legitimate entity.
- It will lead to information disclosure and property damage.
- Large organizations may get trapped in different kinds of scams.

1.2 Purpose :

The focus of this guided project is to apply machine - learning algorithms in order to detect Phishing websites (i.e.) To verify the validity of a website, which will be the most vital objective of our guided project. We implemented classification algorithms and techniques to extract the phishing datasets criteria to classify their legitimacy.

2. Literature Survey:

2.1 Existing Solutions -

Mohammad et al. [1] proposed a model that automatically extracts important features for phishing website detection without requiring any human intervention. Author has concluded in this paper that the process of extracting features by their tool is much faster and more reliable than any manual extraction.

Ahmad et al.[2] proposed three new features to improve accuracy rate for phishing website detection. In this paper, Author used both types of features as commonly known and new features for classification of phishing and
non-phishing sites. At the end the author has concluded this work can be enhanced by using this novel features with decision tree machine learning classifiers.

Mustafa et al.[3] developed a safer framework for detecting phishing websites. They have extracted URL features of websites and used subset based selection techniques to obtain better accuracy .In this paper, the author evaluated CFS subset based and content based subset selection methods And Machine learning algorithms are used for classification purposes.

Chunlin et al. [4] proposed an approach that primarily focuses on character frequency features. In this they have combined statistical analysis of URL with machine learning technique to get a result that is more accurate for classification of malicious URLs. Also they have compared six machine-learning algorithms to verify the effectiveness of the proposed algorithm which gives 99.7% precision with false positive rate less than 0.4%.

Pradeepthi et al.[5] In this paper ,Author studied different classification algorithms and concluded that tree-based classifiers are best and give better accuracy for phishing URL detection. Also

Author uses various Volume 3, Issue 7, September-October-2018 | http:// ijsrcseit.com Purvi Pujara et al. Int J S Res CSE & IT. 2018 September-October-2018; 3(7) : 395-399 398 features such as lexical features, URL based feature, network based features and domain based feature.

2.2 References -

[1] Rami M. Mohammad, Fadi Thabtah, Lee McCluskey: An Assessment of Features Related to Phishing Websites using an Automated Technique:In The 7th International Conference for Internet Technology and Secured Transactions,IEEE,2012

[2] Ahmad Abunadi, Anazida Zainal ,Oluwatobi Akanb: Feature Extraction Process: A Phishing Detection Approach :In IEEE,2013.

[3] Mustafa AYDIN, Nazife BAYKAL : Feature Extraction and Classification Phishing Websites Based on URL : IEEE,2015

[4] Chunlin Liu, Bo Lang : Finding effective classifier for malicious URL detection : InACM,2018

[5] Pradeepthi. K V and Kannan. A: Performance Study of Classification Techniques for Phishing URL Detection: In 2014 Sixth International Conference on Advanced Computing(ICoAC) IEEE,2014

2.2 Problem Statement Definition -

There are e-banking websites that request the users to provide more sensitive information such as credit card details, password etc., for malicious reasons. These websites that mimic trustful URLs and webpages are known as phishing websites. Common causes for web phishing attacks involve:

- Users lack of security awareness
- Not performing sufficient due diligence
- Low-cost phishing and ransomware tools are easy to get hold of
- Malware is becoming more sophisticated and so on.

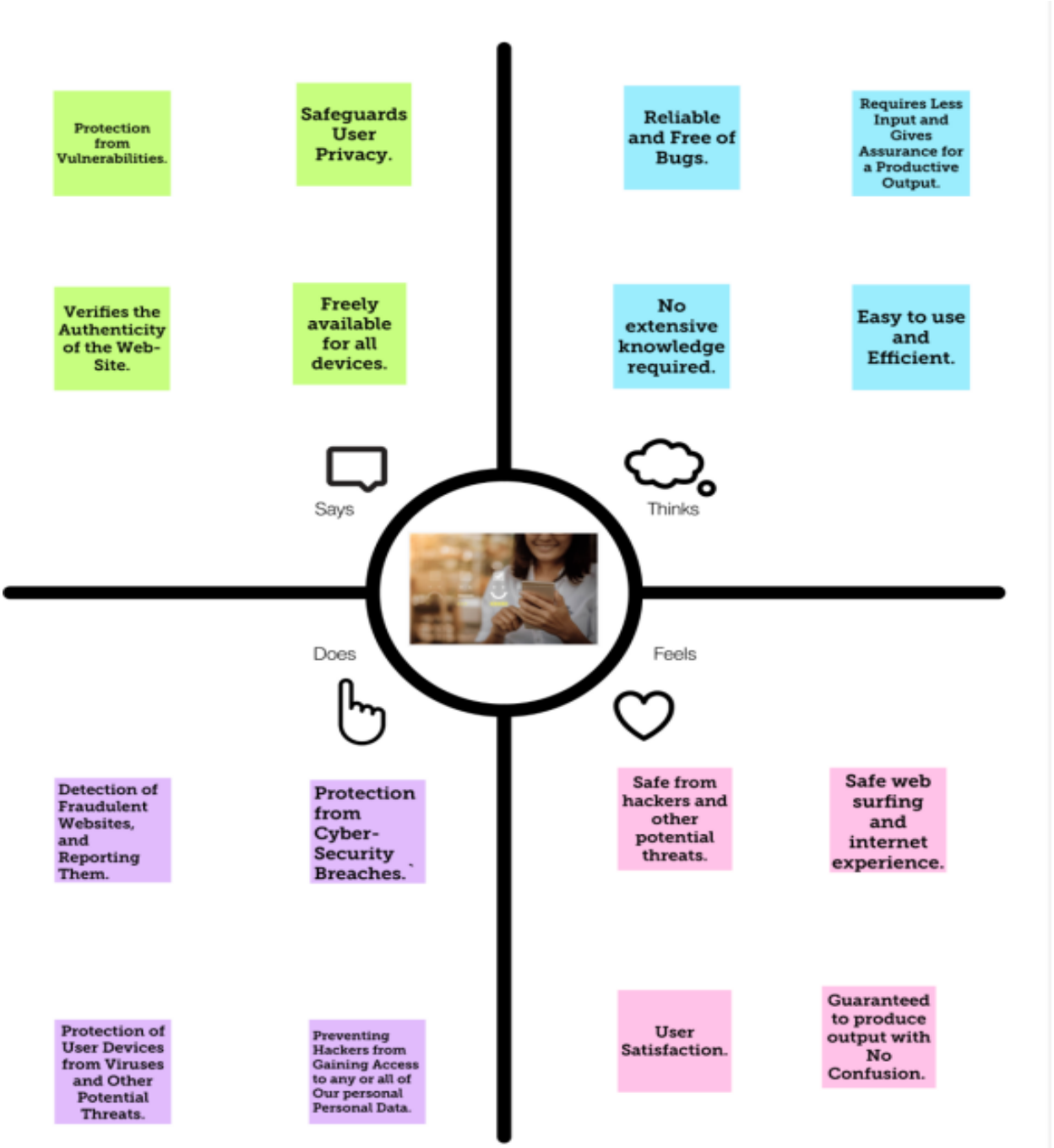
Web phishing is considered to be a threat in various aspects of security on the internet, which might involve scams and private information disclosure. Some of the common threats of web phishing are:

- Attempt to fraudulently solicit personal information from an individual or organization.
- Attempt to deliver malicious software by posing as a trustworthy organization or entity.
- Installing those malwares infects the data that cause a data breach or even nature's forces that takes down your company's data headquarters, disrupting access.

For this purpose, the objective of our project involves building an efficient and intelligent system to detect such websites by applying a machine-learning algorithm which implements classification algorithms and techniques to extract the phishing datasets criteria to classify their legitimacy and as a result of which whenever a user makes a transaction online and makes payment through an e-banking website our system will use a data mining algorithm to detect whether the e-banking website is a phishing website or not.

3. Ideation and Proposed Solution:

3.1 Empathy Map -



3.2 Ideation and Brainstorming -

2

Brainstorm

Writing down ideas that come to mind while addressing the problem statement.

10 minutes

Muazzam N Alseri

Phishing Detection must be Platform - Independent.	Create awareness on the do's and dont's while surfing the internet.	Must be open - source so that other programmers can improve the program by adding additional features or fixing it.
Must be Free and Effective	Must report Fradulent Links	Do not download anything from links obtained through unknown emails or untrusted websites.
Check for viruses.	Add a link scanner.	Train phishing detection machine using data sets.

Mohamed Suhaib Ahmed

Use two-factor authentication to login into your various private accounts on the internet	Do not click on ads or other spam emails containing suspicious links.	Use a trusted ad-blocker service to block out suspicious advertisements and other content on websites.
Be very Cautious with websites that ask about your transaction information or personal details.	Visit trusted domains and use proper and trusted web browsers only.	Cookies must be enabled only in safe and trusted websites.
Malicious websites must be added to blocklist after detection.	Add a domain name scanner and an URL scanner.	Finding an algorithm to train the detection machine.

Dilip Kumar K

Visit only Trusted Websites.	Only websites that appear on top of the search list must be used.	Use filters to clear spam E-mails.
Never enter used-id or passwords in websites other than the ones you have registered.	Do not trust ads on the website.	Try to block out all phishing sites that are previously recognised as such.
Check for fake IP addresses.	Clean data-sets and train the detection machine accordingly.	Count the number of special characters in the URL.

Kishore G

Only open links sent by trusted parties.	Educate others to be cautious by creating awareness	Create awareness on cyber-security attacks and other dangers present in the internet.
Never login any websites while using unknown public networks.	Avoid sharing personal info to anyone unknown	Privacy must be handled appropriately.
Check if the website is already present in blocked list of websites	Count the characters in URL.	Check for poor URL.

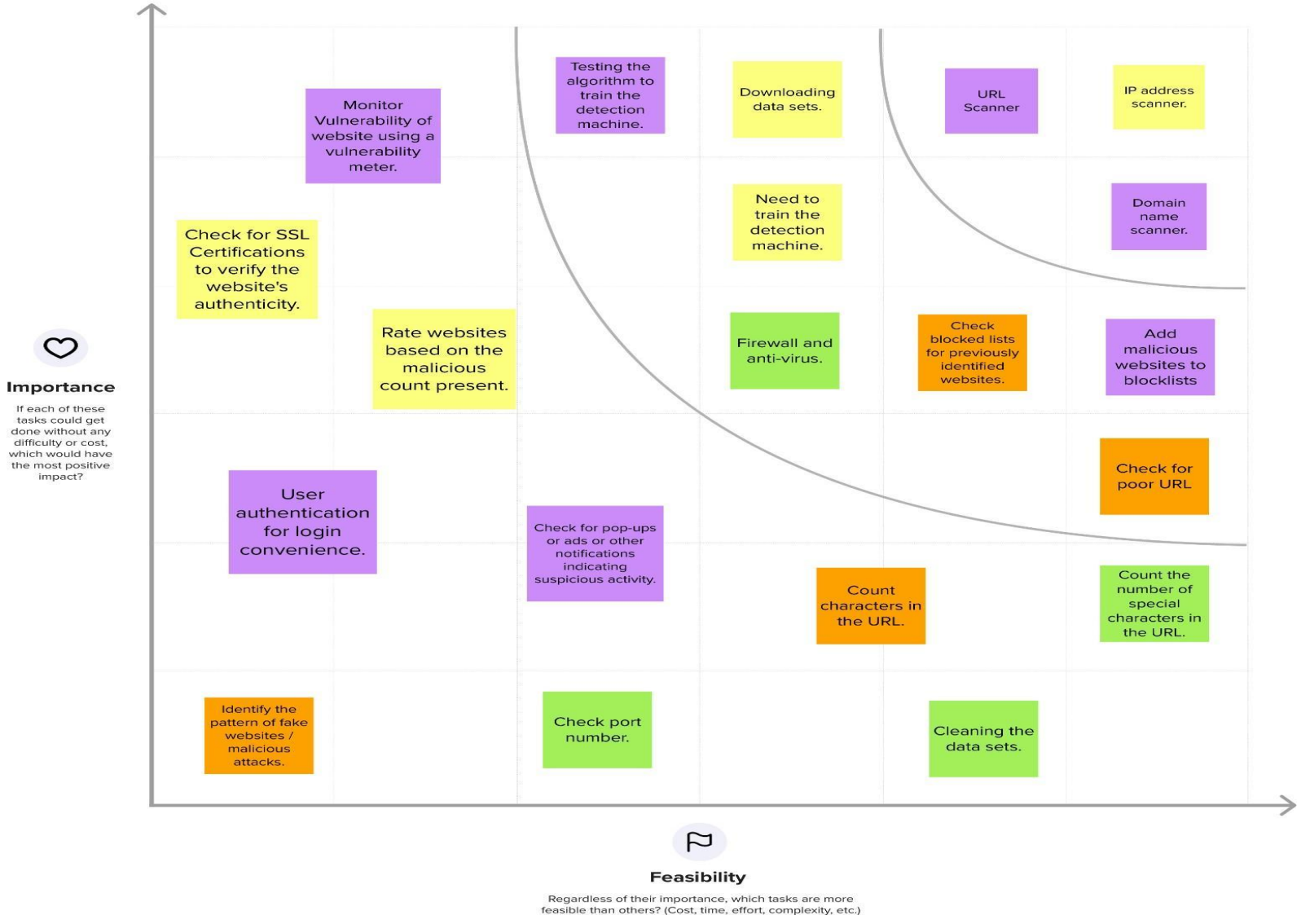
ze

3

Prioritize

It is important for all the members of the team to be on the same page about what's important moving forward. For this, we place all ideas on this grid to determine which ideas are important and which are feasible.

20 minutes



3.3 Proposed Solution -

S. No:	Parameter	Description
1.	Problem Statement (Problem to be solved)	Phishing sites are malicious websites that seem like reliable websites, but they try to steal visitors' personal information such as their username, password, credit account number etc..
2.	Idea / Solution description	Web scraping is what we do with the user's submitted URLs. We then include it into our categorisation machine learning model-based algorithm. This warns the user if our algorithm suggests that the discovered website will not be accurate and that it may perhaps be a phishing site.
3.	Novelty / Uniqueness	Can detect hidden phishing technologies present in websites
4.	Social Impact / Customer Satisfaction	Prevent scams and other malicious intents of using other's personal data
5.	Business Model (Revenue Model)	This model can be implemented alongside other apps and services as add-on for revenue. It can also be used as a plugin for users that surf the internet unprotected
6.	Scalability of the Solution	Can be implemented in all internet browsers as means of countering web phishing on a large scale.

3.4 Problem Solution Fit -

Project Title: Web Phishing Detection

Project Design Phase - I : Solution Fit

Team ID: PNT2022TMID27267

Define CS, fit into CC

1. CUSTOMER SEGMENT(S)

- Users who purchase products online and make payments through e-banking.
- Online payment services
- Web Browsers and Hand-Held applications

6. CUSTOMER CONSTRAINTS

CC

- They won't be able to understand the true nature of the site because they can't observe the transaction site's primary procedure.
- Customers are unable to distinguish between legitimate and fraudulent websites. They are unable to determine whether they should believe the information provided on the websites.

5. AVAILABLE SOLUTIONS

AS

- To identify phishing websites, there are numerous websites that offer phishing detection services. Our phishing detection website's main benefit is that it accurately identifies phishing websites and alerts users before sending them there right away.
- The aforementioned methods evaluate whether the website is included among reputable websites, but they have restrictions on things like exact names and the frequency of additions to the list.

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS

J&P

- To Ensure user safety by preventing user data from being stolen.
- Educate the user on suspicious activity on the surface of the website
- Help the user identify authentic websites from fake phishing ones.

9. PROBLEM ROOT CAUSE

RC

- The issue is that phoney websites can steal client information because of this vulnerability. In order to access the customer's bank account and steal the money, these websites will use the customer's information.
- . The average person won't be very knowledgeable in this field. Even using the web service is more difficult for them.

7. BEHAVIOUR

BE

- Customers utilise phishing detection websites to avoid accessing fraudulent websites and safeguard their personal information on those websites.
- Even if a website appears to be legitimate, users should not believe it.

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS

TR

- Attractive Advertisements / Sales Coupon
- Pop-ups of various kinds
- Links pretending to be legitimate that prompt the user to enter his/her personal details

4. EMOTIONS: BEFORE / AFTER

BEFORE:

Fear, Confused, Threatened, Anxious, Violated..

AFTER:

- Safe, Aware, Confident, Happy

10. YOUR SOLUTION

SL

Using phishing detection websites to stop their information from being leaked is the greatest way to stop clients from visiting fraudulent websites.

8. CHANNELS OF BEHAVIOUR

CH

8.1 ONLINE

Customers utilise phishing websites to prevent the leakage of the information they would otherwise supply to the website.

8.2 OFFLINE

There won't be any issues while the customer is offline because they are unable to access any websites when offline

Identify strong TR & EM

4. Requirement Analysis:

4.1 Functional Requirement -

Following are the Functional Requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Input Validation	The User inputs the URL of the suspicious website to check for its validation.
FR-2	Evaluation	The Model evaluates the website using Blacklist and Whitelist approach
FR-3	Extraction	It retrieves features based on heuristics and visual similarities.
FR - 4	Prediction	The URL is predicted by the model using Machine Learning methods such as Logistic Regression and KNN.
FR - 5	Real Time monitoring	The use of Extension plugin should provide a warning pop-up when they visit a website that it is phished. Extension plugin will have the capability to also detect latest and new phishing websites

FR - 6	Announcement of Events	Model then displays whether the website is a valid site or a phishing site.

4.2 Non - Functional Requirements -

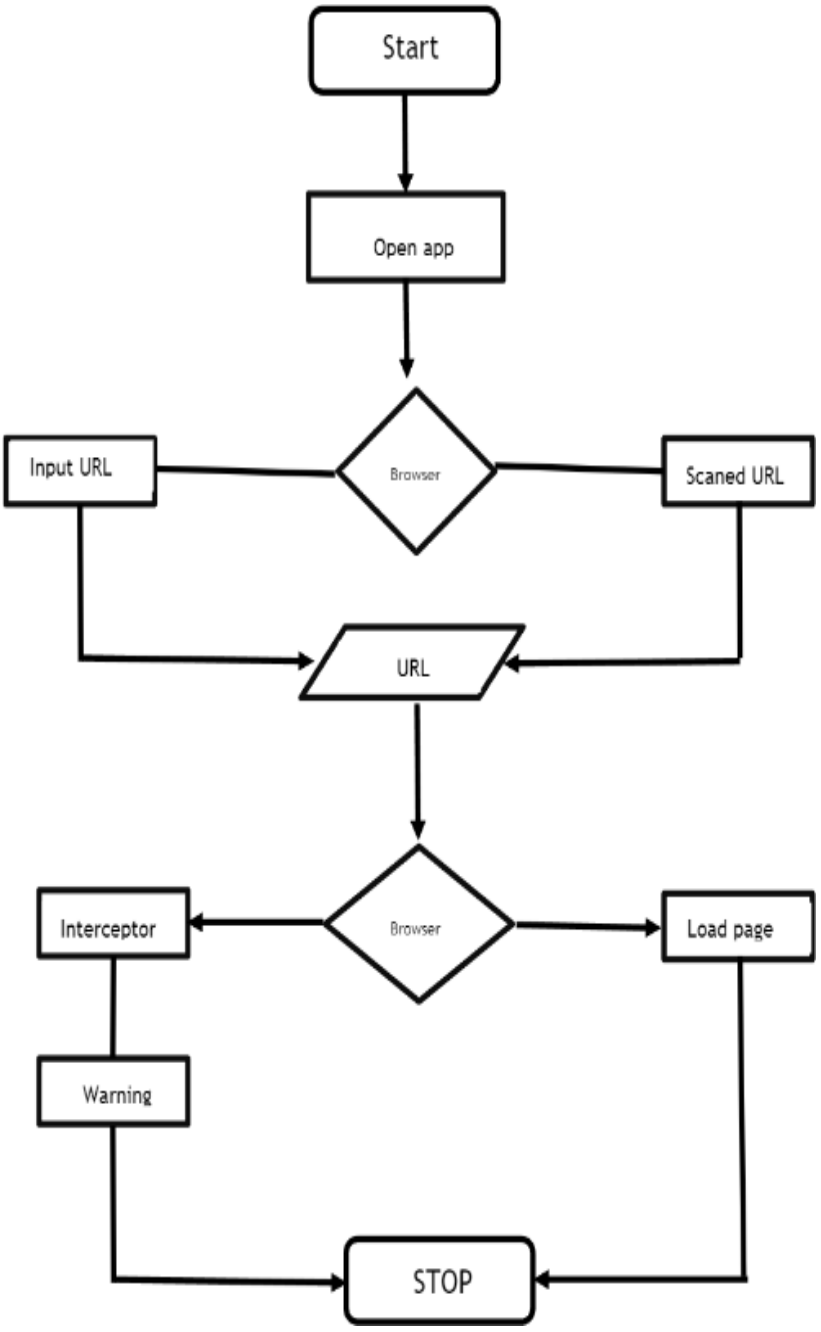
Following are the Non - Functional Requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Responsive UI / UX Design so that users can easily configure the settings based on their personal preference.
NFR-2	Security	Implementation of Updated security algorithms and techniques.
NFR-3	Reliability	It specifies the likelihood that the system or its component will operate without failure for a specified amount of time under prescribed conditions.
NFR-4	Performance	It is concerned with a measurement of the system's reaction time under various load circumstances.
NFR-5	Availability	It represents the likelihood that a user will be able to access the system at a certain moment in time. While it can be represented as an expected proportion of successful requests, it can also be defined as a percentage of time the system is operational within a certain time period.
NFR-6	Scalability	It has access to the highest workloads that will allow the system to satisfy the performance criteria. There are two techniques to enable the system to grow as workloads increase: Vertical and horizontal scaling.

5. Project Design:

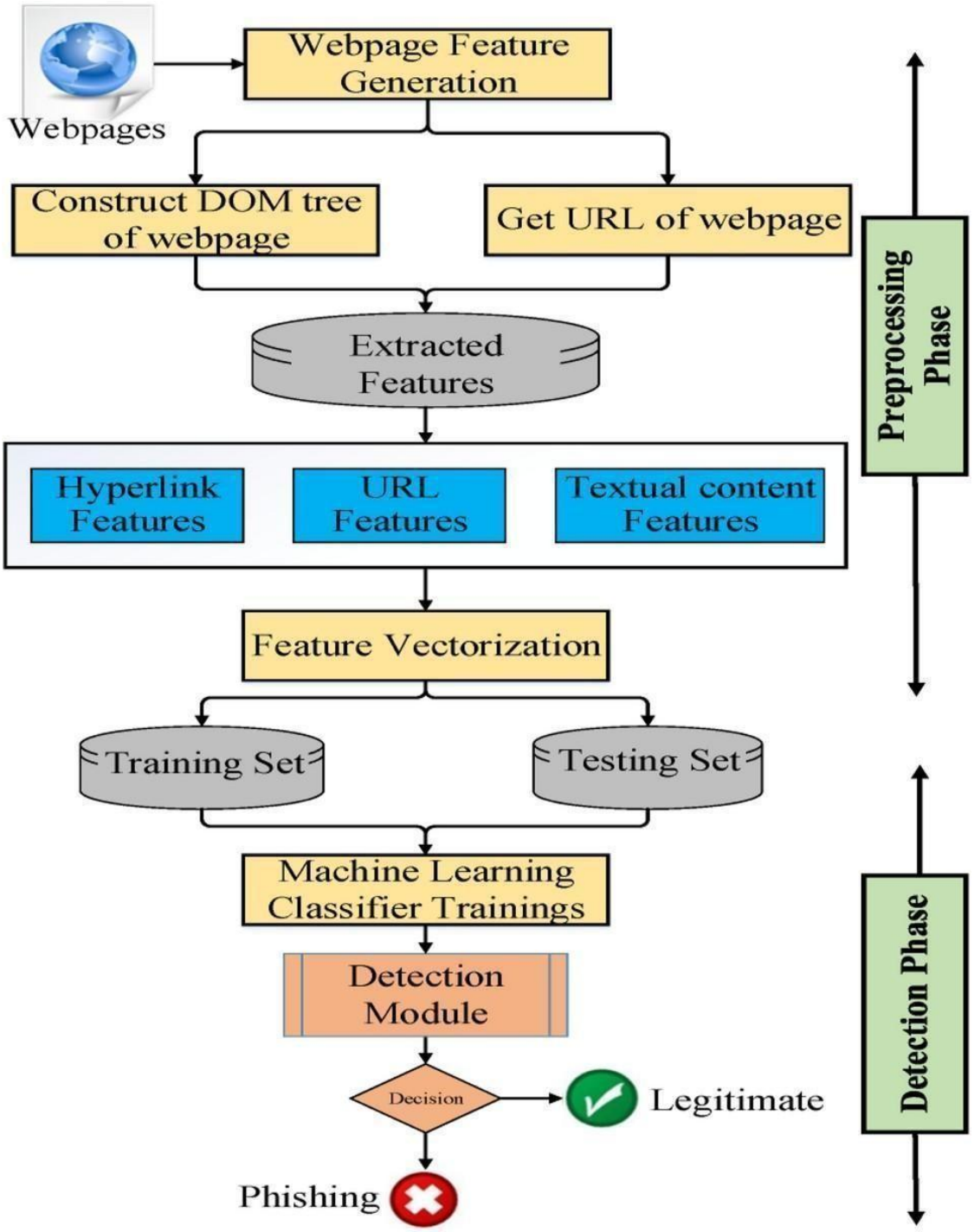
5.1 Data Flow Diagrams -

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

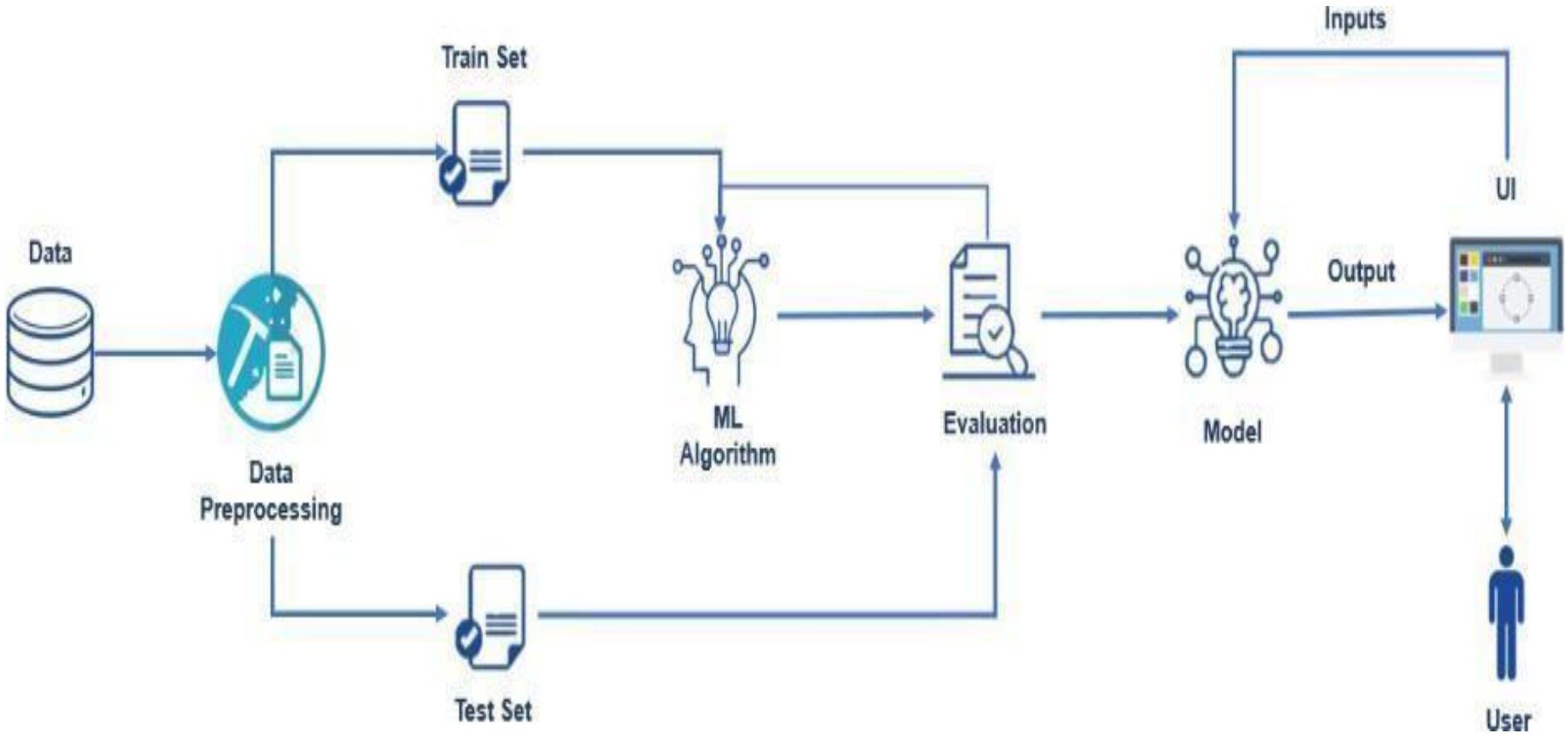


5.2 Solution and Technical Architecture -

- Solution Architecture :



- Technical Architecture :



5.3 User Stories -

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web user)	Login	USN-1	As a user, I can navigate into the website	I can access the page	High	Sprint-1
	Dashboard	USN-2	As a user, I will paste the URL that needs to be checked if it's a phishing website or not	I can paste the URL in the text box	High	Sprint-2
		USN-3	As a user, I can see the output	I can see if it's a safe site	High	Sprint-3

Administrator		USN-4	If a new URL is found, I can add the new state into the database	I can add the new URL	Medium	Sprint-4

6. Project Planning and Scheduling :

6.1 Sprint Planning and Estimation -

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority
Sprint-1	User input	USN-1	User inputs the URL of the suspicious website in the required field to check for its validation.	10	High
Sprint-1	Website Comparison	USN-2	The Model compares the websites using the Blacklist and the Whitelist approach.	10	High
Sprint-2	Feature Extraction	USN-3	After comparison, if nothing is found, then it extracts features using heuristic and visual similarity.	10	Medium

Sprint-2	Prediction	USN-4	The Model predicts the website's URL using Machine learning algorithms such as logistic Regression, Decision Tree and KNN etc..	10	High

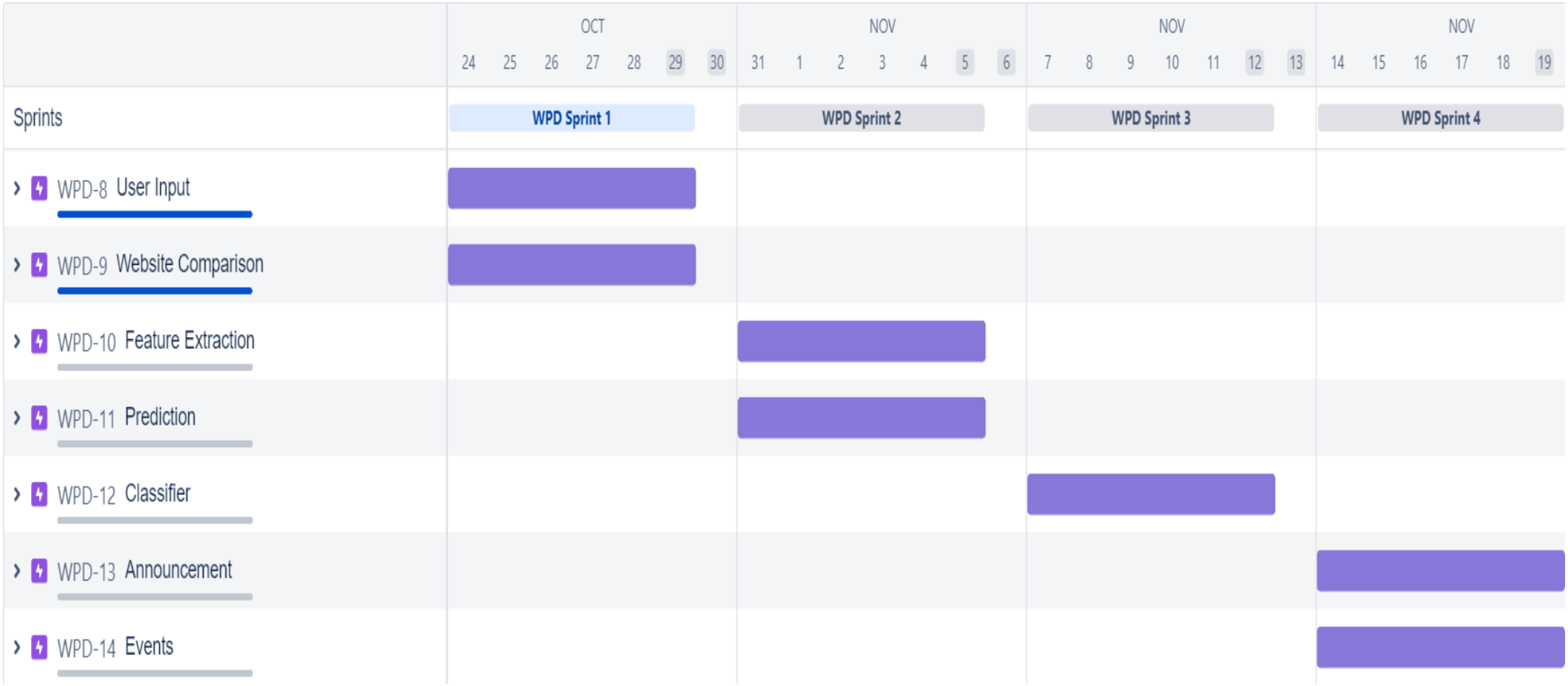
Sprint-3	Classifier	USN-5	The Model sends all of the output to the classifier and it produces the final result.	20	High
Sprint-4	Announcement	USN-6	The Model then displays whether the website is a valid and certified one or a phishing one.	10	High
Sprint-4	Event s	USN-7	This model should have capability of retrieving and displaying accurate results for a website.	10	Mediu m

6.2 Sprint Delivery Schedule -

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)

Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	01 Nov 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	07 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

6.3 JIRA Reports -



7. Coding and Solution :

7.1 Flask App -

```
#importing required libraries

import numpy as np
from flask import Flask, request, jsonify,
render_template import pickle
import pandas as pd

from sklearn import metrics
import warnings

#importing inputScript file which is used to analyze the URL

import inputScript
warnings.filterwarnings('ignore')
from feature import
FeatureExtraction file =
open("model.pkl","rb")
gbc =
pickle.load(file
) file.close()

#loading model
app = Flask(_name_)
@app.route("/",
methods=["GET", "POST"]) def
index():
    if request.method == "POST":
```

```
        url = request.form["url"]
        obj =
        FeatureExtraction(url)
        x = np.array(obj.getFeaturesList()).reshape(1,30)
        y_pred
        =gbc.predict(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 = "It is {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(debug=True,port=
    2 002)
```

7.2 Building HTML Page -

HTML:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <!-- meta tags-->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <!-- Css Attachment-->
  <link rel="stylesheet" type="text/css" href="c.css">

  <title>Home</title>
</head>
<body>

  <!--Header starts -->
  <header class="header" id="home">
    <h1 class="logo"><a href="#"> WEB PHISHING DETECTION</a></h1>
    <ul class="main-nav">
      <li><a href="#">Home</a></li>
      <li><a href="#about">About</a></li>

    </ul>
  </header>

  <section class="about">
    <div class="main">
```

```
        <div class="about-text">
          <h1>Solution for web phishing attacks</h1>
          <p>Beware of phishing websites that are taking your sensitive
          information and login passwords.
          </p>
          <div class="search-box">
            <input class="search-txt" type="text" name="" placeholder="Enter your
            link">
            <i class="fas fa-search"></i>
          </div>
             <button type="button">Check URL</button>


        </div>
        <img src="Cyber-News-Rundown_PhishingEmail_800x4001.gif"
        alt="Personal Image">
      </div>
    </section>
    <!--Header ends -->


    <!--About starts -->
    <div class="container" id="about">
      <h2 class="title" id="h2">ABOUT</h2>
      <div class="text-area">
        <div class="text1">
          <p>Phishing is a form of fraud in which the attacker tries to
learn sensitive information such as login credentials or account
information by sending as a reputable entity or person in email or other
communication channels. Phishing is popular among attackers, since it
is easier to trick someone into clicking a malicious link which seems
legitimate than trying to break through a computer's defense systems.
The malicious links within the body of the message are designed to
make it appear that they go to the spoofed organization using that
```

```
organization’s logos and other legitimate contents.</p>
        </div>
        <div class="text2">
          <p>In several fields, automation has been achieved through the extensive use of machine learning. Our approach is based on the aggregate analysis method to automatically develop rules to determine layout similarity of web sites and then detect phishing pages. Researchers also utilize machine learning to detect phishing assaults based on numerous aspects. Our strategy is divided into two parts. It leverages the attributes of the website layout to first train a similarity classifier, which is then used to identify phishing pages.
          </p>
      </div>
      </div>
   </div>
   <!--About ends -->
</html>
```

CSS:

```
html {
   box-sizing:
   border-box
   ; margin: 0;
   padding: 0;
   font-family: sans-serif;
  }
  body {
   line-height
   : 1.6;
   min-height:
```

```
  100vh;
 }

 /* ==================================

  Header Starts
 ========================================== */
 ul {
  margin
  : 0;
  padding: 0;
  list-style: none;
 }
.header {
  flex-direction:
  row;
  justify-content:
  space-between;
  position:
  sticky; top:
  0px;
 } .header a {
  text-decoration
  : none; color:
  #2d5faf;
 }
```

```
.logo {
margin
: 0;
font-size: 1.45em;
}

.main-nav {
margin-top:
5px;

}

.logo a,
.main-nav a {
margin: 0
5px; padding:
6px 15px;
text-transform
: uppercase;
text-align: center;
display: block;
}
.main-nav a
{ color:
```

```
#2d5faf;
font-size
:
.99em;
}

.main-nav a:hover
{ border: 2px solid
#9ddfed;
border-radius:
13%;
}

.header {
overflow
: hidden;
padding-top:
.5em;
padding-bottom:
.5em; border: 1px
solid #a2a2a2;
background-color
: #fff;
}
.about
```

```
width: 100%;
padding: 0;
background: linear-gradient(-45deg,
#ee7752,#e73c7e,#23a6d5,#23d5ab); background-size:
400% 400%;
position: relative;
}

.about img {
height:450px;
width: 650px;
padding: 25px
0px;
}

.about-text
{ width:
550px;
color:
#fff;
font-family: sans-serif;
}

.about-tex
t h1 {
line-height:
```

```
#ffffff;
font-size:
50px;
text-transform:
capitalize;
margin-bottom:
20px;
}

.main {
height:
550px;
width:
100%;
max-width:
100%;
display: flex;
align-items: center;
justify-content: space-around;
}

.about-text p
{ color:
#d9d9d9;
font-size:
20px;
letter-spacing:
```

2px;

```
  line-height
  : 35px;
  margin-bottom: 30px;
  }

 search-box{
  position:
  absolute; top:
  50%;
  left: 50%;
  transform:
  translate(0%,0%)
  ; background:
  #2f3640; height:
  40px;
  border-radius:
  40px; padding:
  10px;

}
.search-box >
  .search-txt{
  width: 240px;
  padding: 0
  6px; top:
  50px;
```

```
}
.search-box >
  .search-btn{
  background
  : white;
   }
.search-btn
  { color:
  #185ae8;
  float: right;
  width:
  40px;
  height
  :
  40px;
  border-radius:
  50%;
  background:
  #eeeff1;
  display: flex;
  justify-content
  : center;
  align-items:
  center;
  transition:
```

```
  0.4s;
  text-decoration: none;
}
```

```
.search-txt{
border: none;
background:
none; outline:
none; float:
left; padding:
0;
color:
white;
font-size:
16px;
transition
: 0.4s;
line-height:
40px; width:
0px;
}

 button {
  background:
  #9ddfed;
  color:
  #fafafa;
  border: 2px
  solid
  transparent;
```

text-decoration

: none;

```
font-weight:
bold;
font-size
:
16px; padding:
8px 20px;
border-radius
:
30px;
transition:
.3s;
}

button:hover {
background:
transparent; border:
2px solid #9ddfed;
cursor: pointer;
}

/* ====================================
Header Ends
==================================== */

/* ====================================
```

```
About Starts
========================================== */


.container

{

 width: 100%;
padding: 20px 0;

}


.container h2

 {

margin-top:

4rem;

font-size:

42px;

text-align:

center; color:

#2d5faf;

text-transform

: capitalize;

text-overflow

: hidden;

}


.text-area {

height:
```

```
200px;
width:
100%;
max-width: 100%;
display: flex;
align-items: center;
justify-content: space-around;
}
.text1,
.text2 {
padding: 10px
30px;
letter-spacing:
0.5px;
font-size
:
17px; color:
#575252;
}
```

```
@media (min-width: 769px) {

 .header,
 .main-n
 a v {
 display:
 flex;

 }

 .header {
 flex-direction:column;
 }
}
@media (min-width: 1025px) {
 .header {
 flex-direction: row;
 justify-content: space-between;
 }

}
```

7.3 Integrating Flask with Scoring Endpoint -

23 lines (18 sloc) | 1.36 KB

Raw Blame

```python
import requests
import json
# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = "oMYpPuqEigKw47IHZhLRljBu0Vw_djzFNYwlI8dZlLRF"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey": API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}

# NOTE: manually define and pass the array(s) of values to be scored in the next line
payload_scoring = {"input_data": [{"field": [["f0","f1","f2","f3","f4","f5","f6","f7","f8","f9","f10","f11","f12","f13","f14","f15","f16","f17","f18","f19","f20

response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/5db35193-0eb1-4c56-be21-745a459ba937/predictions?version=2022-11-17', json=
print("Scoring response")
print(response_scoring.json())

predictions=response_scoring.json()

pred=print(predictions['predictions'][0]['values'][0][0])
if(pred != 1):
        print("This is a Legitimate Website.")
else:
        print("This is a fake phishing website")
```

8. Testing :

8.1 Test Cases -

Test case ID	Feature Type	Component	Test Scenario	Prerequisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y / N)	BUG ID	Executed By
LoginPage_TC_OO 1	Functional	Home Page	Verify user is able to see the Landing Page when user can type the URL in the box		1.Enter URL and click go; 2.Type the URL; 3.Verify whether it is processing or not.	https://phishing-shield.herokuapp.com/	Should Display the Webpage	Working as expected	Pass		N		Dilip Kumar K
LoginPage_TC_OO 2	UI	Home Page	Verify the UI elements is Responsive		1. Enter URL and click go 2. Type or copy paste the URL 3. Check whether the button is responsive or not 4. Reload and Test Simultaneously	https://phishing-shield.herokuapp.com/	Should Wait for Response and then gets Acknowledge	Working as expected	Pass		N		Kishore G
LoginPage_TC_OO 3	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://phishing-shield.herokuapp.com/	User should observe whether the website is legitimate or not.	Working as expected	Pass		N		Muazzam N Alseri
LoginPage_TC_OO 4	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.	https://phishing-shield.herokuapp.com/	Application should show that Safe Webpage or Unsafe.	Working as expected	Pass		N		Mohamed Suhaib Ahmed

LoginPage TC OO 5	Functional	Home Page	Testing the website with multiple URLs		1. Enter URL (https://phishing-shield.herokuapp.com /) and click go 2. Type or copy paste the URL to test 3. Check the website is legitimate or not 4. Continue if the website is secure or be cautious if it is not secure	1. https://www.google.com/ 2. delgets.com	User can able to identify the websites whether it is secure or not	Working as expected	Pass		N		Muazzam N Alseri

8.2 User Acceptance Testing -

1. Defect Analysis:

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	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	77

2. Test Case Analysis:

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51
Security	2	0	0	2

Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

9. Results:

9.1 Performance of the Model -

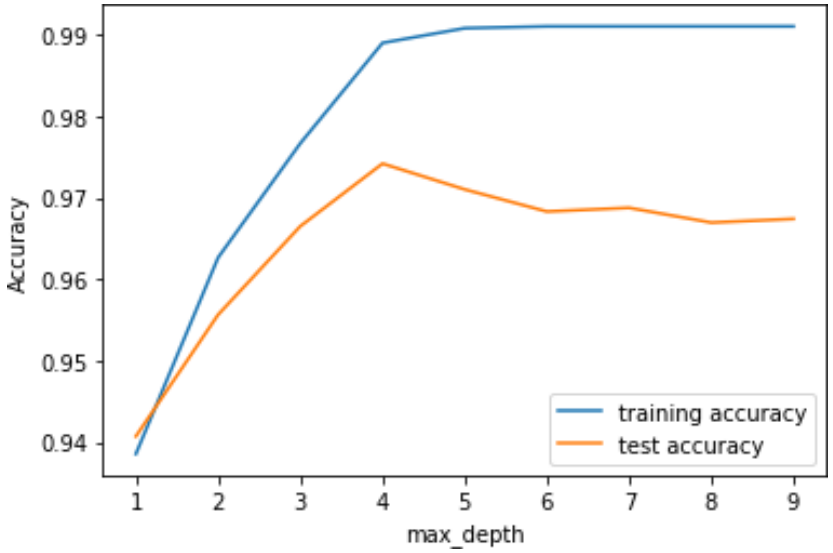
S.No.	Parameter	Values	Screenshot

1.	Metrics	**Classification Model:**
		
		Gradient Boosting Classification -
		
		Accuracy Score = **97%**
			`#computing the classification report of the model`
			
			`print(metrics.classification_report(y_test, y_test_gbc))`
			
			precision recall f1-score support
			
			-1 0.99 0.96 0.97 976
			1 0.97 0.99 0.98 1235
			
			accuracy 0.97 2211
			macro avg 0.98 0.97 0.97 2211
			weighted avg 0.97 0.97 0.97 2211

1.	Tune the Model	**Hyperparameter Tuning :**
		
		Validation Method - KFOLD and Cross Validation Method
		
		Accuracy Score = **95%**
			In [78]:
			`#KFOLD and Cross Validation Model`
			
			`from scipy.stats import wilcoxon`
			`from sklearn.datasets import load_iris`
			`from sklearn.ensemble import GradientBoostingClassifier`
			`from xgboost import XGBClassifier`
			`from sklearn.model_selection import cross_val_score, KFold`
			
			`# Load the dataset`
			`X = load_iris().data`
			`y = load_iris().target`
			
			`# Prepare models and select your CV method`
			`model1 = GradientBoostingClassifier(n_estimators=100)`
			`model2 = XGBClassifier(n_estimators=100)`
			`kf = KFold(n_splits=20, random_state=None)`
			`# Extract results for each model on the same folds`
			`results_model1 = cross_val_score(model1, X, y, cv=kf)`
			`results_model2 = cross_val_score(model2, X, y, cv=kf)`
			`stat, p = wilcoxon(results_model1, results_model2, zero_method='zsplit');`
			`stat`
			
			Out[78]: 95.0

1. Metrics Classifications Report:

Performance -



Out[83]:

	ML Model	Accuracy	f1_score	Recall	Precision
0	Gradient Boosting Classifier	0.974	0.977	0.994	0.986
1	CatBoost Classifier	0.972	0.975	0.994	0.989
2	Random Forest	0.969	0.972	0.992	0.991
3	Support Vector Machine	0.964	0.968	0.980	0.965
4	Decision Tree	0.958	0.962	0.991	0.993
5	K-Nearest Neighbors	0.956	0.961	0.991	0.989
6	Logistic Regression	0.934	0.941	0.943	0.927
7	Naive Bayes Classifier	0.605	0.454	0.292	0.997
8	XGBoost Classifier	0.548	0.548	0.993	0.984
9	Multi-layer Perceptron	0.543	0.543	0.989	0.983

2. Tuning The Model [Hyper - Tuning] :

```
In [58]: #HYPERPARAMETER TUNING
         grid.fit(X_train, y_train)
```

Out[58]:

GridSearchCV

```
GridSearchCV(cv=5,
             estimator=GradientBoostingClassifier(learning_rate=0.7,
                                                  max_depth=4),
             param_grid={'max_features': array([1, 2, 3, 4, 5]),
                         'n_estimators': array([ 10,  20,  30,  40,  50,  60,  70,  80,  90, 100, 110, 120, 130,
       140, 150, 160, 170, 180, 190, 200])})
```

estimator: GradientBoostingClassifier

```
GradientBoostingClassifier(learning_rate=0.7, max_depth=4)
```

GradientBoostingClassifier

```
GradientBoostingClassifier(learning_rate=0.7, max_depth=4)
```

```
In [59]: print("The best parameters are %s with a score of %0.2f"
               % (grid.best_params_, grid.best_score_))
```

```
The best parameters are {'max_features': 5, 'n_estimators': 200} with a score of 0.97
```

Validation Methods [KFOLD and Cross Folding] :

In [78]:
```
#KFOLD and Cross Validation Model

from scipy.stats import wilcoxon
from sklearn.datasets import load_iris
from sklearn.ensemble import GradientBoostingClassifier
from xgboost import XGBClassifier
from sklearn.model_selection import cross_val_score, KFold

# Load the dataset
X = load_iris().data
y = load_iris().target

# Prepare models and select your CV method
model1 = GradientBoostingClassifier(n_estimators=100)
model2 = XGBClassifier(n_estimators=100)
kf = KFold(n_splits=20, random_state=None)
# Extract results for each model on the same folds
results_model1 = cross_val_score(model1, X, y, cv=kf)
results_model2 = cross_val_score(model2, X, y, cv=kf)
stat, p = wilcoxon(results_model1, results_model2, zero_method='zsplit');
stat
```

Out[78]:
```
95.0
```

5x2CV combined F test

In [89]:
```
from mlxtend.evaluate import combined_ftest_5x2cv
from sklearn.tree import DecisionTreeClassifier, ExtraTreeClassifier
from sklearn.ensemble import GradientBoostingClassifier
from mlxtend.data import iris_data

# Prepare data and clfs
X, y = iris_data()
clf1 = GradientBoostingClassifier()
clf2 = DecisionTreeClassifier()

# Calculate p-value
f, p = combined_ftest_5x2cv(estimator1=clf1,
                            estimator2=clf2,
                            X=X, y=y,
                            random_seed=1)

print('f-value:', f)
print('p-value:', p)
```

```
f-value: 1.727272727272733
p-value: 0.2840135734291782
```

10. Advantages and Disadvantages :

Advantages -

1. Accurate and Efficient.
2. Gives a clear idea for the user to stay away from potentially malicious websites.
3. Does not require installation of any third - party softwares to detect phishing websites from valid ones.
4. Cost Effective and Consumes less time.

Disadvantages -

1. Using some features might prove to be time consuming.
2. False Alarms and Less Accurate Results.
3. Phishers can sometimes bypass the filter database by transforming words.
4. Needs constant learning and training of data.

11. Conclusion :

It is found that phishing attacks are very crucial and it is important for us to get a mechanism to detect it. As very important and personal information of the user can be leaked through phishing websites, it becomes more critical to take care of this issue. This problem can be easily solved by using any of the machine learning algorithms with the classifier. We already have classifiers which give a good prediction rate of the phishing. We have seen that the existing systems give less accuracy so we proposed a new phishing method that employs URL based features and also we generated classifiers through several machine learning algorithms.

We have found that our system provides us with 93% of accuracy for Logistic Regression Classification method, 95.4% of accuracy for K - Nearest Neighbour Classifier, 96% of accuracy for Support Vector Machine Classifier, 95.6% of accuracy for Decision Tree Classifier, 96.5% of accuracy for Random Forest Classifier and finally 97% of accuracy when using Gradient Boosting Classifier.Hence we found that the best among all the above classifiers is Gradient Boosting Classifier which shows maximum accuracy. Therefore, The proposed technique in this guided project is much more secure as it detects new and previous phishing sites.

12. Future Scope :

In the future, if we get structured datasets of phishing data, then we can perform phishing detection much faster and with more efficiency with accurate results than any other technique, as we can use a combination of any other two or more classifiers to get maximum accuracy. We also plan to explore various phishing techniques that use Lexical features, Network based features,Content based features, Webpage based features and HTML and JavaScript features of web pages which can improve the performance of the system. In particular, we extract features from URLs and pass it through the various classifiers.

- ***Github Link*** :

  https://github.com/IBM-EPBL/IBM-Project-20179-1659714221.git

- ***Video Demonstration Link :***

  https://youtu.be/f5lxCFdfPYI