WEB PHISING DETECTECTION

TEAM ID : PNT2022TMID34845

PROJECT REPORT

Submitted by

ABINAYA R (962819104003)

ABITHA VINCY S (962819104004)

JAYYENU JR (962819104044)

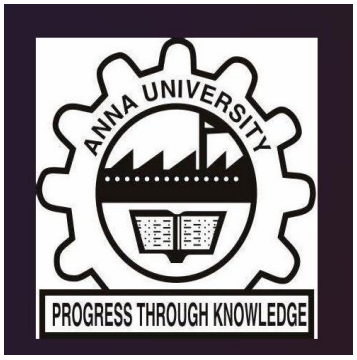
JEYA SELSHIA J (96281919104049)

IN PARTIAL FULFILLMENT FOR THE AWARD OF DEGREE OF

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING



INDEX

1.INTRODUCTION

The Internet has become an indispensable part of our life, However, It also has provided opportunities to anonymously perform malicious activities like Phishing. Phishers try to deceive their victims by social engineering or creating mock up websites to steal information such as account ID, username, password from individuals and organizations. Although many methods have been proposed to detect phishing websites.

Phishers have evolved their methods to escape from these detection methods. One of the most successful methods for detecting these malicious activities can be done using Machine Learning Techniques. This is because most Phishing attacks have some common characteristics which can be identified by machine learning methods.

1.1 PROJECT OVERVIEW:

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:

- 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.

In order to detect and predict e-banking phishing websites, we proposed an intelligent, flexible and effective system that is based on using classification algorithms. We implemented classification algorithms and techniques to extract the phishing datasets criteria to classify their legitimacy. The e-banking phishing website can be detected based on some important characteristics like URL and domain identity, and security and encryption criteria in the final phishing detection rate. Once a user makes a transaction online when he 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.

1.2 PURPOSE

There is a significant chance of exploitation of user information. For these reasons, phishing in modern society is highly urgent, challenging, and overly critical. There have been several recent studies against phishing based on the characteristics of a domain, such as website URLs, website content, incorporating both the website URLs and content, the source code of the website and the screenshot of the website. However, there is a lack of useful anti-phishing tools to detect malicious URL in an organization to protect its users. In the event of malicious code being implanted on the website, hackers may steal user information and install malware, which poses a serious risk to cybersecurity and user privacy. Malicious URLs on the Internet can be easily identified by analysing it through Machine Learning (ML) technique. The conventional URL detection approach is based on a blacklist (set of malicious URLs) obtained by user reports or manual opinions.

On the one hand, the blacklist is used to verify an URL and on the other hand the URL in the blacklist is updated, frequently. However, the numbers of malicious URLs not on the blacklist are increasing significantly. For instance, cybercriminals can use a Domain Generation Algorithm (DGA) to circumvent the blacklist by creating new malicious URLs. Thus, an exhaustive blacklist of malicious URLs is almost impossible to identify the malicious URLs. Thus new malicious URLs cannot be identified with the existing approaches. Researchers suggested methods based on the learning of computer to identify malicious URLs to resolve the limitations of the system based on the blacklist.

Malicious URL detection is considered a binary classification task with two-class predictions: malicious and benign. The training of the ML method consists of finding the best mapping between the d-dimensional vector space and the output variable. This strategy has a strong generalization capacity to find unknown malicious URLs compared to the blacklist approach.

2.LITERATURE SURVEY

Jason Hong(2009) [4] Phishing attacks are a significant security threat to users of the Internet, causing tremendous economic loss every year. Past work in academia has not been adopted by industry in part due to concerns about liability over false positives. However, blacklist-based methods heavily used in industry are slow in responding to new phish attacks, and tend to be easily overwhelmed by phishing techniques. Phishing has become a substantial threat for internet users and a major cause of financial losses. In these attacks the cybercriminals carry out user credential information and users

can fall victim. The current solution againstphishing attacks are not sufficient to detect and work against novel phishes. This paper presents a systematic review of the previous and current research waves done on Internet.

Hussain Ahmed, et.al (2007) [3] Malicious URLs are harmful to every aspect of computer users. Detecting of the malicious URL is very important. Currently, detection of malicious web pages techniques includes blacklist and white-list methodology and machine learning classification algorithms are used. However, the blacklist and white-list technology is useless if a particular URL is not in list. In This paper, we propose a multi-layer model for detecting malicious URL.

JunHo Huh (2013) [6] We propose a new phishing detection heuristic based on the search results returned from popular web search engines such as Google, Bing and Yahoo. The full URL of a website a user intends to access is used as the search string, and the number of results returned and ranking of the website are used for classification.

Dr. Gunikhan Sonowal (2017) [1] Phishing remains a basic security issue in cyberspace. In phishing, assailants steal sensitive information from victims by providing a fake site which looks like the visual clone of a legitimate site. Phishing shall be handled using various approaches. It is established that single filter methods would be insufficient to detect different categories of phishing attempts.

Rami Mustafa (2007) [7] Phishing is described as the art of emulating a website of a creditable firm intending to grab user's private information such as usernames, passwords and social security number. Phishing websites comprise a variety ofcues within its content-parts as well as browser-based security indicators. Several solutions have been proposed to tackle phishing.

Shubhangi Wankhede (2004) [9] Detecting any Phishing site is extremely an intricate and dynamic issue including numerous variables and criteria. Due to the ambiguities associated with phishing location, fluffy information mining procedures can be a viable instrument in detecting phishing websites. In this paper, we propose a strategy which consolidates fluffy rationale alongside information digging algorithms for detecting phishing websites.

Ankit singh (2007) [8] Phishing emails are more dynamic and cause high risk of significant data, brand and financial loss to average computer user and organizations. To address this problem, we propose a hybrid feature selectionapproach based on combination of content-based and behaviour-based.

Our proposed hybrid features selections are able to achieve 93% accuracy rate as compared to other approaches. In addition, we successfully tested the quality of our proposed behaviour -based feature using the Information Gain, Gain Ratio and Symmetrical Uncertainty.

Adwan Yaseen (2014) [5] Phishing attacks are one of the trending cyber-attacks that apply socially engineered messages that are communicated to people from professional hackers aiming at fooling users to reveal their sensitive information, the most popular communication channel to those messages is through users' emails. This paper presents an intelligent classification model for detecting phishing emails using knowledge discovery, data mining.

Andrew H. Sung (2010) [10] Phising has become an important cybersecurity problem. The centralized blacklist approach used by most web browsers usually fails to detect zero-day attacks, leaving the ordinary users vulnerable to new phishing schemes; therefore, learning machine based approaches have been implemented for phishing detection. Many existing techniques in phishing website detection seem to include as many features as can be conceived, while identifying a relevant and representative subset of features to construct an accurate classifier remains an interesting issue in this particular application of machine learning.

Hiba Zuhair (2007) [2] Web services motivate phishers to evolve more deceptive websites as their never-ending threats to users. This intricate challenge enforces researchers to develop more proficient phishing detection approaches that incorporate hybrid features, machine learning classifiers, and feature selection methods. However, these detection approaches remain incompetent in classification performance over the vast web. This is attributed to the limited selection of the best features from the massive number of hybrid ones, and to the variant outcomes of applied feature selection methods in the realistic condition. In this topic, this paper surveys prominent researches, highlights their limitations, and emphasises on how they could be improved to escalate detection performance. This survey restates additional peculiarities to promote certain facets of the current research trend with the hope to help researchers on how to develop detection approaches and obtain the best quality outcomes of feature selection.

2.1 EXISTING PROBLEMS:

Nowadays Phishing becomes a main area of concern for security researchers because it is not difficult to create the fake website which looks so close to legitimate website. Experts can identify fake websites but not all the users can identify the fake website and such users become the victim of phishing attack. Main aim of the attacker is to steal banks account credentials. Phishing attacks are becoming successful because lack of user awareness. Since phishing attack exploits the weaknesses found in users, it is very difficult to mitigate them but it is very important to enhance phishing detection techniques. Phishing may be a style of broad extortion that happens once a pernicious web site act sort of a real one memory that the last word objective to accumulate unstable info, as an example, passwords, account focal points, or MasterCard numbers. all the same, the means that there square measure some of contrary to phishing programmingand techniques for recognizing potential phishing tries in messages and characteristic phishing substance on locales, phishes think about new and crossbreed procedures to bypass the open programming and frameworks.

Phishing may be a fraud framework that uses a mixture of social designing what is additional, advancement to sensitive and personal data, as an example, passwords associate degree open-end credit unpretentious elements by presumptuous the highlights of a reliable individual or business in electronic correspondence. Phishing makes use of parody messages that square measure created to seem substantial and instructed to start out from true blue sources like money connected institutions, online business goals, etc, to draw in customers to go to phony destinations through joins gave within the phishing websites.

2.2 REFERENCES:

[1] Dr. Gunikhan Sonowal: 'Phishing Scams Cost American Businesses Half A Billion Dollars A Year'. Forbes, 5 May 2017. Accessed Jan 2018.

[2] Hiba Zuhair 'Phishing and Pharming – The Deadly Duo'. SANS Institute, 2007. Accessed Jan 2018.

[3] Hussain Ahmed, Riaz Khan . 'Online frauds in banks with phishing'. The Journal of Internet Banking and Commerce, vol.12, no.2, pp.1–27, 2007.

[4] Hong, J. 'The Current State of Phishing Attacks'. Communication of the ACM, vol.55, no.1, pp.74– 81, 2012.

[5] Adwan Yaseen 'Classification of Phishing Email Using Random Forest Machine Learning Technique'. Journal of Applied Mathematics, vol.2014, pp.1–7, Apr 2014.

[6] Jun Ho Huh, 'Spear-phishing: how to spot and mitigate the menace'. Computer Fraud & Security, Jan 2013, pp.11–16. Accessed Jan 2018.

[7] Rami Mustafa 'Social Phishing'. In Communications of the ACM 50, no.10 (2007): 94–100.

[8] Akit Singh: 'Protecting People from Phishing: the design and evaluation of an embedded training email system'. In Proceedings of the SIGCHI conference on Human Factors in Computing Systems, pp.905–914. ACM, 2007.

[9] Shubhangi Wankhede: Protecting (even) naive web users from spoofing and phishing attacks'. Bar Ilan University Technical Report, 2004.

[10] Andrew H. Sung ‘Who Falls for Phish? A Demographic Analysis of Phishing Susceptibility and Effectiveness of Interventions’. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp.373–382. ACM, 2010

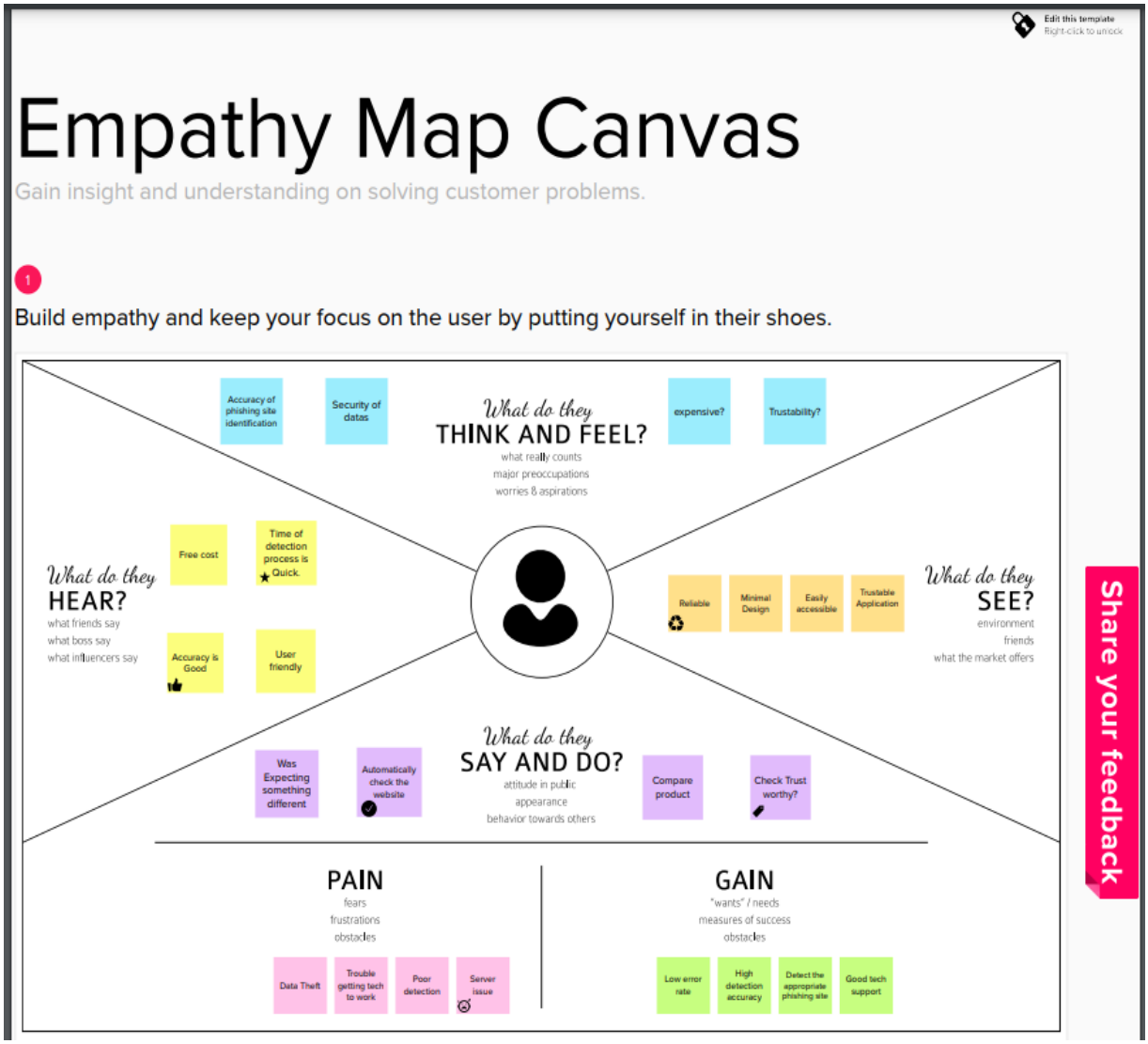
2.3 PROBLEM STATEMENT DEFINITION:

Internet has dominated the world by dragging half of the world’s population exponentially into the cyber world. With the booming of internet transactions, cybercrimes rapidly increased and with anonymity presented by the internet, Hackers attempt to trap the end-users through various forms such as phishing, SQL injection, malware, man-in-the-middle, domain name system tunnelling, ransomware, web trojan, and so on. Among all these attacks, phishing reports to be the most deceiving attack.

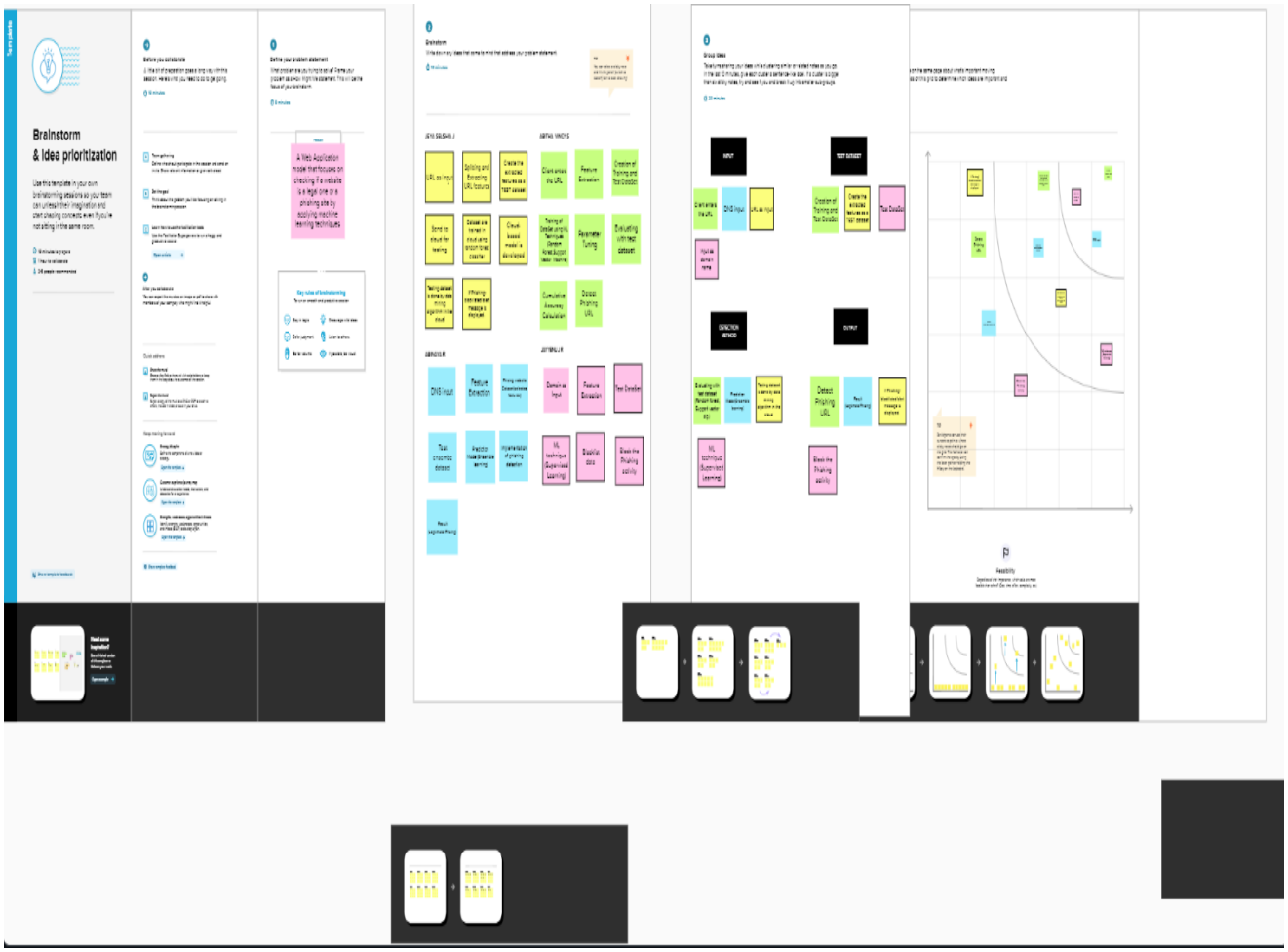
Our main aim of this paper is classification of a phishing website with the aid of various machine learning techniques to achieve maximum accuracy and concise model.

3.IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 IDEATION AND BRAINSTORMING



3.3 PROPOSED SOLUTION

Proposed Solution Template:

Project team shall fill the following information in proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	A phishing website is a fake website, with domain similar in name and appearance to an official website. They're made in order to fool someone into believing it is legitimate. And then extract login credentials or confidential information such as credit card details from victims to perform malicious activities.
2.	Idea / Solution description	In order to detect and predict phishing website, we proposed an intelligent, flexible and effective system that is based on using classification Data mining algorithm. We implemented classification algorithm and techniques to extract the phishing data sets criteria (URL and Domain Identity, security and encryption) to classify their legitimacy. Then detect whether the website is a phishing site or not.
3.	Novelty / Uniqueness	One of the major contributions of this project is the selection of different new features, which are capable enough to detect 0-h attacks, and these features do not depend on any third-party services. In particular, we extract character level Term Frequency-Inverse Document Frequency (TF-IDF) features from noisy parts of HTML and plaintext of the given webpage.
4.	Social Impact / Customer Satisfaction	Since Data mining algorithm used in this system, it provides better performance as compared to other traditional classifications algorithms. With the help of this system user can also purchase products online without any hesitation. The accuracy of phishing site identification is around 89%.

3.4 PROBLEM SOLUTION FIT

4. REQUIREMENT ANALYSIS

4.1 FUNTIONAL REQUIREMENTS

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	URL of the website to be checked.	To detect the website for legitimate percentage, the URL of the webpage is required.
FR-2	An active online cloud to DB and model.	To store the blacklisted website and to deploy the ML model online, a cloud storage is required. Also, the URL feature is extracted and processed on the model.
FR-3	Internet Connectivity	As we train and test data set in IBM cloud, internet connectivity is mandatory in-order to access the cloud and test the URL.
FR-4		

4.2 NON FUNTIONAL REQUIREMENTS

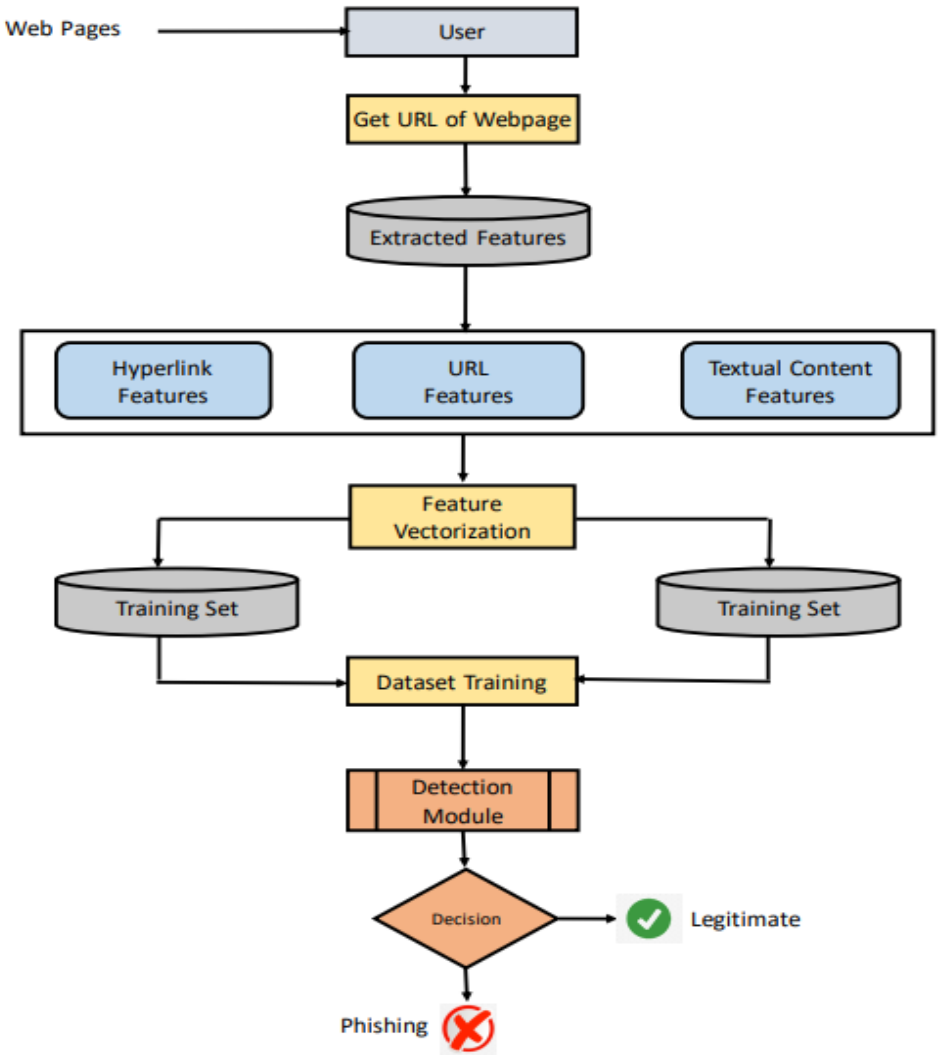
Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Can be used by any payment website or by any users.
NFR-2	**Security**	High level of security as we use ML technology.
NFR-3	**Reliability**	Once the extension is added, the application predicts automatically if the system has internet connectivity.
NFR-4	**Performance**	Prediction rate is 94% higher, when compared to classical methods.
NFR-5	**Availability**	Available once the cloud connectivity is established.
NFR-6	**Scalability**	Can be made compatible with Android application.

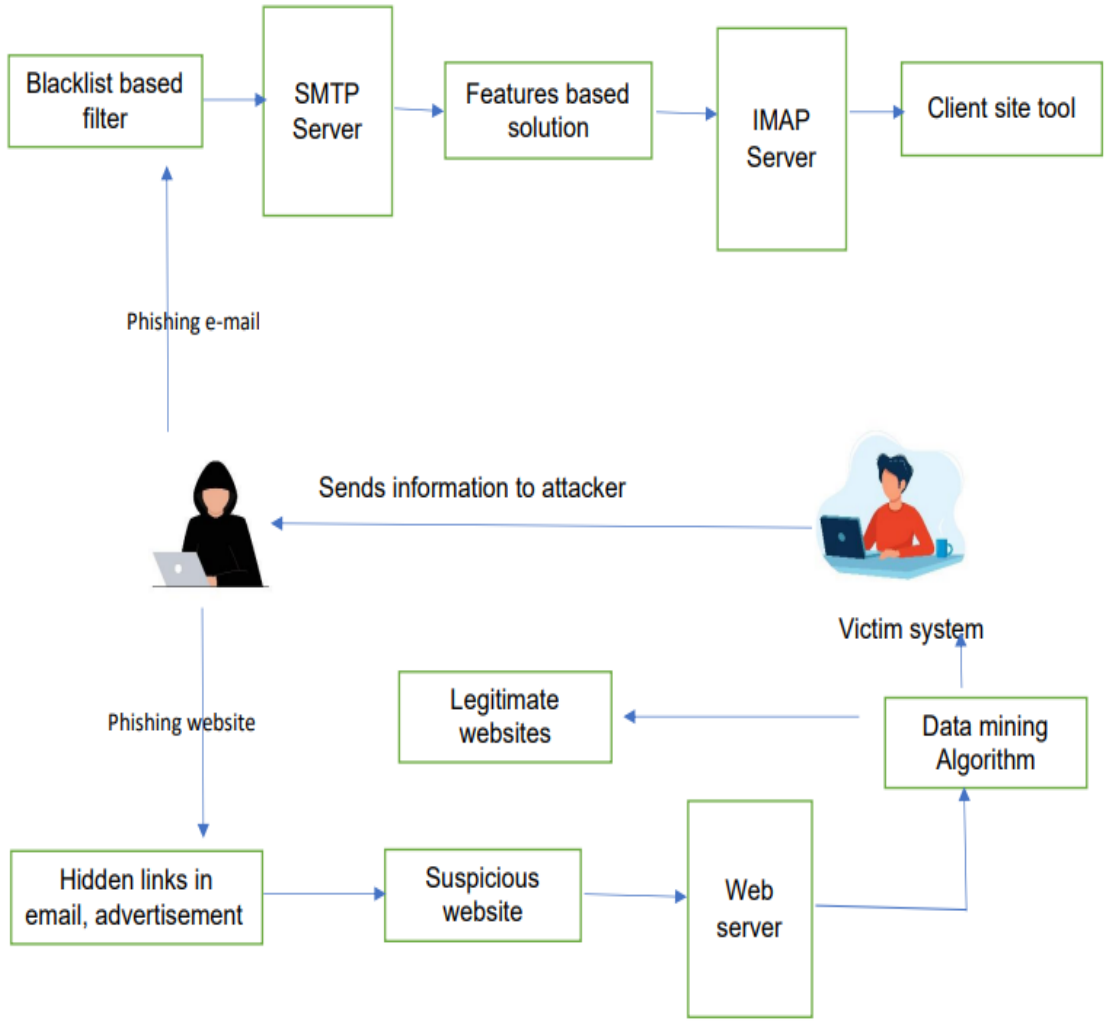
5.PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS:

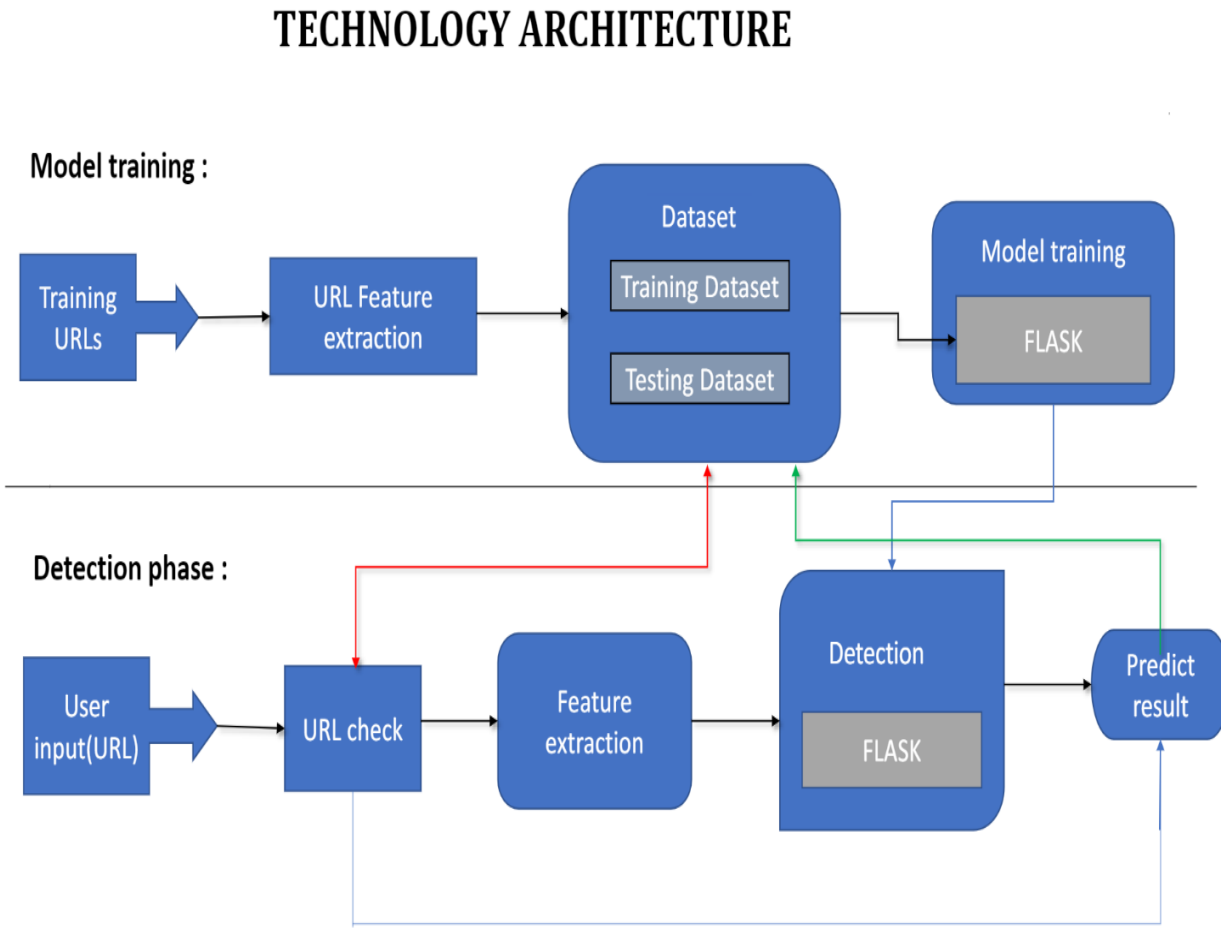


5.2 SOLUTION & TECHNICAL ARCHITECTURE

Solution Architecture Diagram:



TECHNICAL ARCHITECTURE



6.PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Product Backlog, Sprint Schedule, and ser inter-Estimation (4 Marks)

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	HTML Page Creation	USN-1	User interface to receive URL as input.	10	High	Jeya Selshia J Abinaya R
Sprint- 1	Data set collection	USN-2	Collecting dataset to train the model. Dataset has number of parameters considered for classification.	10	High	Abitha Vincy S Jayyenu J R
Sprint-1	URL detection	USN-3	URL is the first thing to analyse a website to decide whether it is a phishing or not	10	High	Jeya Selshia J Abitha Vincy S
Sprint-1		USN-4	Some of URL-Based Features are • Digit count in the URL • Total length of URL • Checking whether the URL is typo-squatted or not • Checking whether it includes a legitimate brand name or not • Number of subdomains in URL	10	High	Abinaya R Jayyenu J R

			• TLD is one of the commonly used one			
Sprint-2	Domain detection	USN-5	The purpose of Phishing Domain Detection 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 to us.	10	High	Jeya Selshia J Abinaya R
Sprint-2		USN-6	Some useful Domain-Based Features are • Its domain name or its IP address in blacklists of well-known reputation services? • How many days passed since the domain was registered? • Is the registrant name hidden?	10	High	Abitha Vincy S Jayyenu J R
Sprint-3	Page based features and Content based features	USN-7	Page-Based Features are using information about pages which are calculated reputation ranking services. Obtaining these types of features requires active scan to target domain. Page contents are processed for us to detect whether target domain is used for phishing or not	10	High	Abinaya R Abitha Vincy S
Sprint-3			• Global pagerank • Country pagerank • Position at the Alexa top 1 million site Some processed information about pages are • Page titles • Meta tags • Hidden text • Text in the body • Images etc.			Jeya Selshia J Jayyenu J R
Sprint-4	Detection process	USN-8	Detecting Phishing Domains is a classification problem, so it means we need labeled data which has samples as phish domains and legitimate domains in the training phase	20		Jeya Selshia J Abinaya R

6.2 SPRINT DELIVERY SECHEDULE

Project Tracker, Velocity & Burndown Chart: (4 Marks)

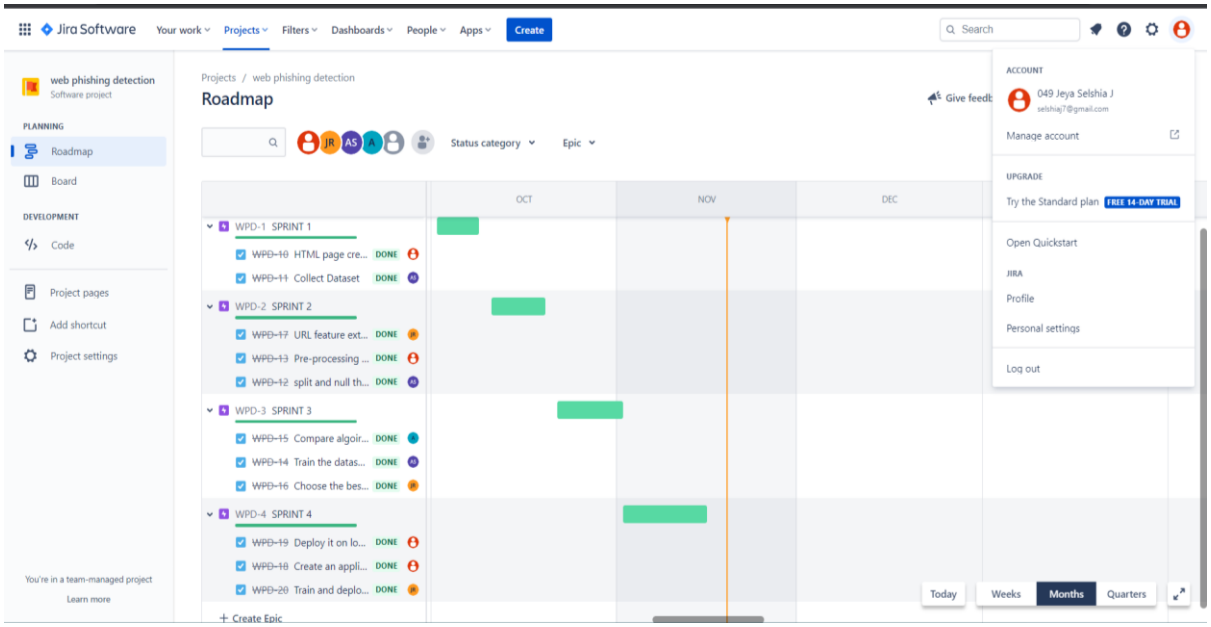
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	4 Days	08 Oct 2022	12 Oct 2022	10	12 Oct 2022
Sprint-2	20	4 Days	17 Oct 2022	21 Oct 2022	10	21 Nov 2022
Sprint-3	20	4 Days	26 Oct 2022	30 Oct 2022	10	30 Oct 2022
Sprint-4	20	4 Days	09 Nov 2022	12 Nov 2022	20	12 Nov 2022

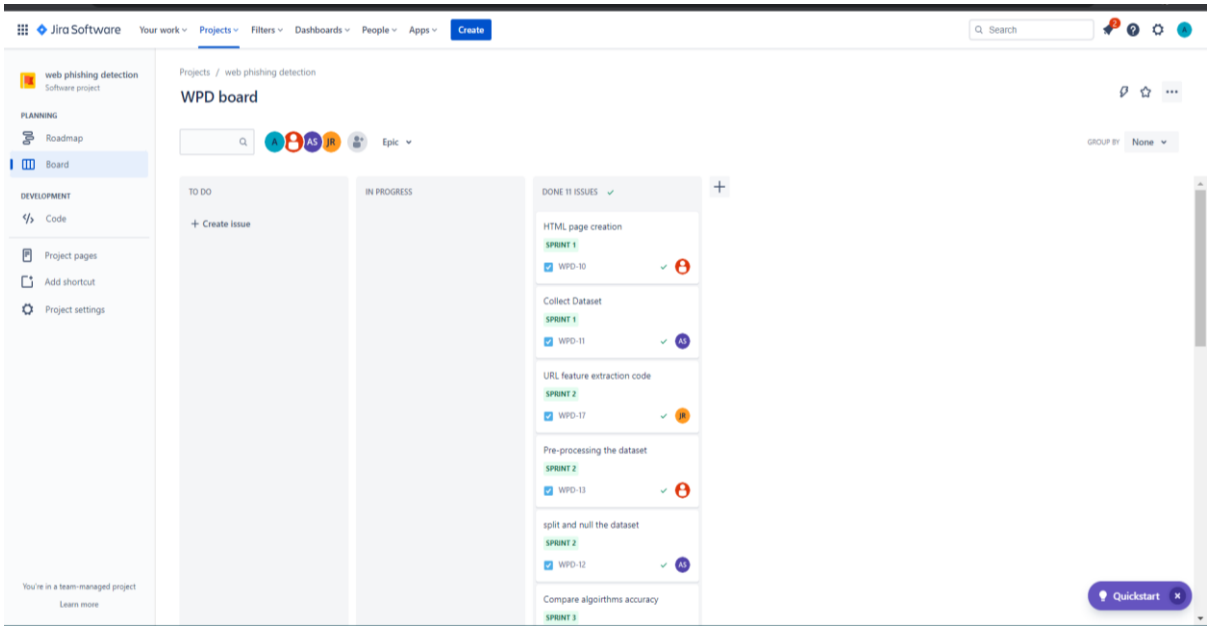
Velocity:

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

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

6.3 REPORTS FROM JIRA





7 CODING AND SOLUTIONING

7.1 Feature 1

User Interface: To receive the URL as input.

HTML page code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <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 develop for identify the safety of url.">
    <meta name="keywords" content="phishing url,phishing,cyber security,machine learning,classifier,python">
    <meta name="author" content="ibm_group_PNT2022TMID34845">

    <!-- BootStrap -->
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css"
        integrity="sha384-9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYxFfc+NcPb1dKGj7Sk" crossorigin="anonymous">
    <link href="static/styles.css" rel="stylesheet">
    <title>URL detection</title>
```

```
</head>
<body>
<div class=" container">


<div class="row">
      <div class="form col-md" id="form1">
         <h2>PHISHING WEBSITE 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" >Click to check</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+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
        crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"
        integrity="sha384-
Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
        crossorigin="anonymous"></script>
    <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"
        integrity="sha384-
OgVRvuATP1z7JjHLkuOU7Xw704+h835Lr+6QL9UvYjZE3Ipu6Tp75j7Bh/kR0JKI"
        crossorigin="anonymous"></script>


    <script>
            let x = '{{xx}}';
            let num = x*100;
            if (0<=x && x<0.50){
                num = 100-num;
            }
            let txtx = num.toString();
```

```
        if(x<=1 && x>=0.50){
            var label = "Website is "+txtx +"% safe to use...";
            document.getElementById("prediction").innerHTML = label;
            document.getElementById("button1").style.display="block";
        }
        else if (0<=x && x<0.50){
            var label = "Website is "+txtx +"% unsafe to use..."
            document.getElementById("prediction").innerHTML = label ;
            document.getElementById("button2").style.display="block";
        }
    </script>


<footer>
    <h>TEAM ID: PNT2022TMID34845</h>

</footer>
</body>

</html>
```

7.2 Feature 2

URL Feature extraction: T o split and extract the url features such as HTTPS, Anchor URL, prefix-suffix etc.

Feature extraction code:

```
import ipaddress
import re
import urllib.request
from bs4 import BeautifulSoup
import socket
import requests
from googlesearch import search
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.domain = ""
        self.whois_response = ""

  self.urlparse = ""
        self.response = ""
        self.soup = ""
```

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

try:
    self.urlparse = urlparse(url)
    self.domain = self.urlparse.netloc
except:
    pass

try:
    self.whois_response = whois.whois(self.domain)
except:
    Pass
 self.features.append(self.UsingIp())
self.features.append(self.longUrl())
self.features.append(self.shortUrl())
self.features.append(self.symbol())
self.features.append(self.redirecting())
self.features.append(self.prefixSuffix())
self.features.append(self.SubDomains())
self.features.append(self.Hppts())
self.features.append(self.DomainRegLen())
```

```
self.features.append(self.Favicon())

self.features.append(self.NonStdPort())
self.features.append(self.HTTPSDomainURL())
self.features.append(self.RequestURL())
self.features.append(self.AnchorURL())
self.features.append(self.LinksInScriptTags())
self.features.append(self.ServerFormHandler())
self.features.append(self.InfoEmail())
self.features.append(self.AbnormalURL())
self.features.append(self.WebsiteForwarding())
self.features.append(self.StatusBarCust())

self.features.append(self.DisableRightClick())
self.features.append(self.UsingPopupWindow())
self.features.append(self.IframeRedirection())
self.features.append(self.AgeofDomain())
self.features.append(self.DNSRecording())
self.features.append(self.WebsiteTraffic())
self.features.append(self.PageRank())
self.features.append(self.GoogleIndex())
self.features.append(self.LinksPointingToPage())
self.features.append(self.StatsReport())
```

```
 # 1.UsingIp
    def UsingIp(self):
        try:
            ipaddress.ip_address(self.url)
            return -1
        except:
            return 1
# 2.longUrl
    def longUrl(self):
        if len(self.url) < 54:
            return 1
        if len(self.url) >= 54 and len(self.url) <= 75:
            return 0
        return -1


    # 3.shortUrl
    def shortUrl(self):
        match =
re.search('bit\.ly|goo\.gl|shorte\.st|go2l\.ink|x\.co|ow\.ly|t\.co|tinyurl|tr\.i
m|is\.gd|cli\.gs|'

'yfrog\.com|migre\.me|ff\.im|tiny\.cc|url4\.eu|twit\.ac|su\.pr|twurl\.nl|snip
url\.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|om\.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|buzurl\.com|cutt\.us|u\.bb|yourls\.org|'

'x\.co|prettylinkpro\.com|scrnch\.me|filoops\.info|vzturl\.com|qr\.net|1url\.com|tweez\.me|v\.gd|tr\.im|link\.zip\.net', self.url)
if match:
            return -1
        return 1
 # 4.Symbol@
    def symbol(self):
        if re.findall("@",self.url):
            return -1
        return 1
    # 5.Redirecting//
    def redirecting(self):
        if self.url.rfind('//')>6:
            return -1
        return 1
.....................................................
def getFeaturesList(self):
        return self.features
```

7.3 MODEL BIULDING AND APPLICATION BUILDING

Creating model and application :

To create a model, the dataset is processed with many ML models and among this a model with high accuracy rate is chosen.This data set comes under classification problem, as the input URL is classified as phishing (1) or legitimate (0)The ML agorithms used to test and train the dataset are:

Decision Tree

Random Forest

XGBoost Classifier

Support Vector Machines

Here, we use XGBoost Classifier to build model as it has 97% accuracy rate when compared.Now, the created model is used to process the input URL.

An application is built using Flask and the model is imported to it.

app.py code:

```
#importing required libraries


from flask import Flask, request, render_template

import numpy as np
```

```
import pandas as pd
from sklearn import metrics

import warnings
import pickle
warnings.filterwarnings('ignore')
from feature import FeatureExtraction

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



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)
```

8.RESULTS

1. In this project various URL features are taken as parameters like

index

having_IP

having_IP_Address

URL_HTTPS

URL_Length

Shortining_Service

having_At_Symbol

double_slash_redirecting

Prefix_Suffix

SSLfinal_State

port

SFH

Abnormal_URL

popUpWidnow

Iframe

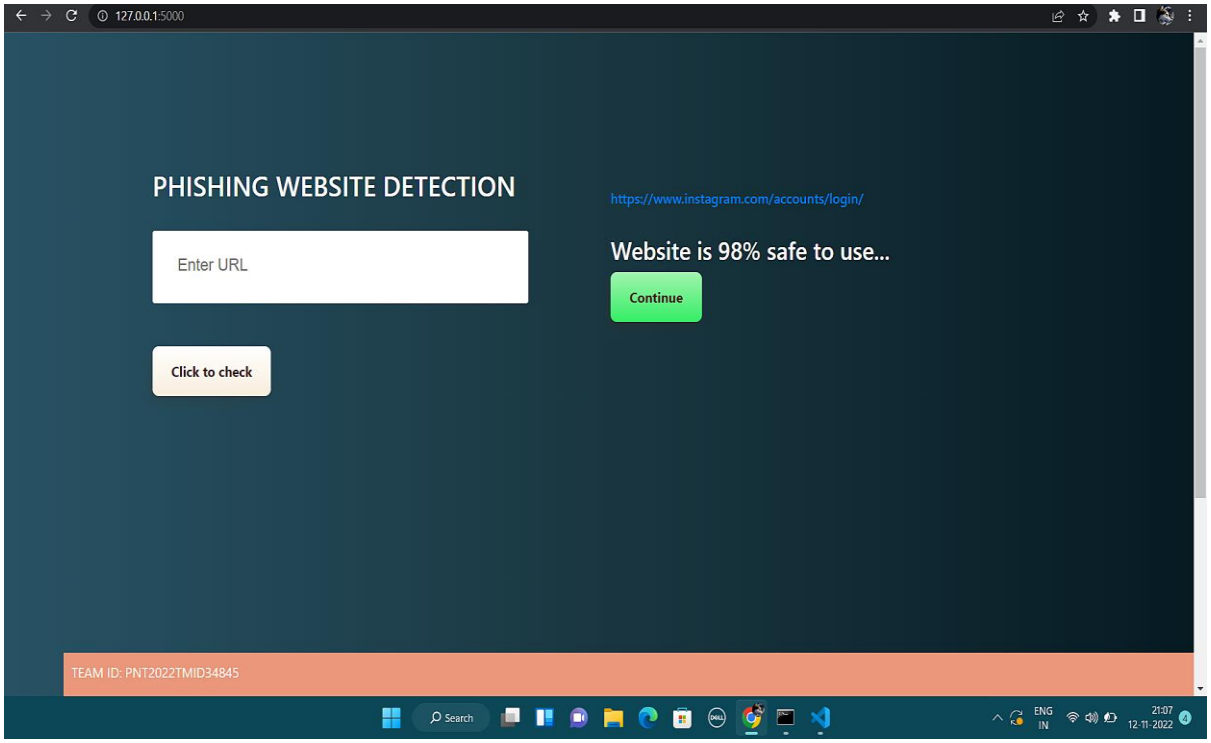
DNSRecord

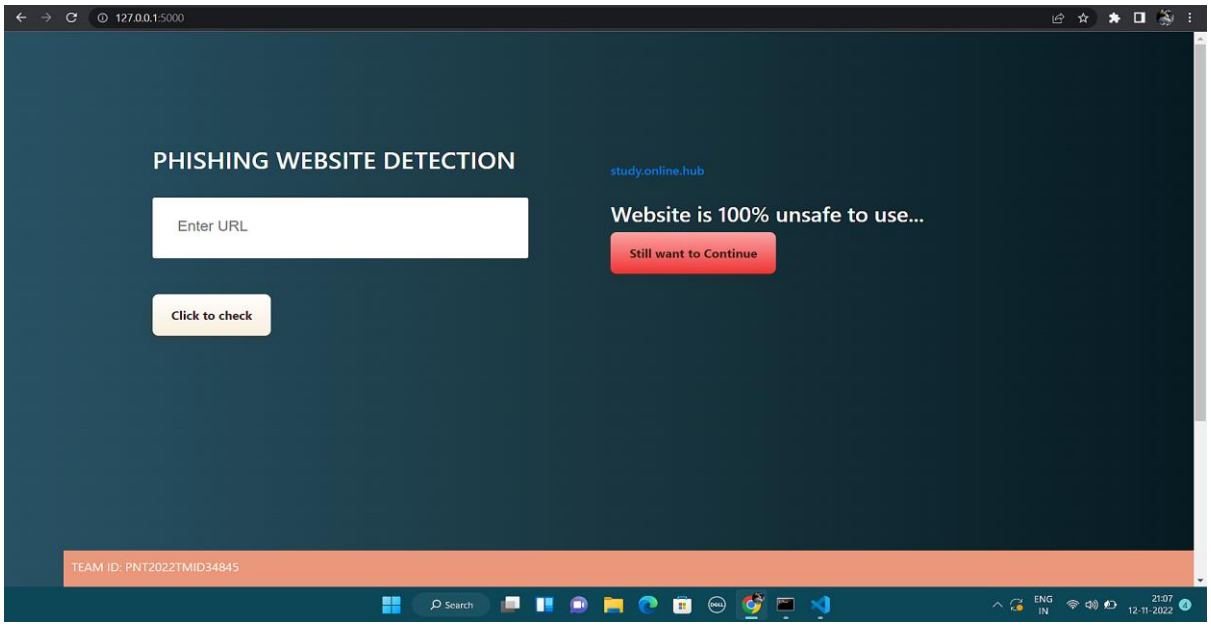
Statistical_report, so that the URL is verified effectively.

2. Models are trained and tested in IBM cloud and Flask is integrated to it.

3. The final conclusion on the Phishing dataset is that the some feature like "HTTTPS", "AnchorURL", "WebsiteTraffic" have more importance to classify URL is phishing URL or not.

4. Gradient Boosting Classifier(XGBoostClassifier) currectly classify URL upto 97.4% respective classes and hence reduces the chance of malicious attachments.


127.0.0.1:5000
PHISHING WEBSITE DETECTION
Enter URL
Click to check
https://www.instagram.com/accounts/login/
Website is 98% safe to use...
Continue
TEAM ID: PNT2022TMID34845


127.0.0.1:5000
PHISHING WEBSITE DETECTION
Enter URL
Click to check
study.online.hub
Website is 100% unsafe to use...
Still want to Continue
TEAM ID: PNT2022TMID34845

9.ADVANTAGE AND DISADVANTAGES:

Advantages :

1.This system can be used by many E-commerce or other websites in order to have good customer relationship.

2.User can make online payment securely without any fear.

3.XGBoostClassifier algorithm used in this system provides better performance in prediction as compared to other traditional classifications algorithms.

Disadvantages:

1.If Internet connection fails, this system won't work.

2.URL are to be given manually with consumes time.

10.CONCLUSION

In this project, the user should provide the URL of the website which he is about to access. Then the system will process the URL and detect if the website is legitimate or not. The system will show the legitimate percent of the website given.Thus, making the web surfing safe.

11.FUTURE

This web application can be further developed to create a chrome extension , so as when ever a user explore a website a notification indicating if the website is safe or malicious can be popped automatically. Thus reducing the fraud effectively.

Also it can be further developed to block the access to illegitimate website.

12.APPENDIX

Source Code Github Link:

https://github.com/IBM-EPBL/IBM-Project-33992-1660230287

Project Demo Link:

https://youtu.be/_CpsWFE3ozY