Project Report

WEB PHISHING DETECTION

SUBMITTED BY :

TEAM ID : **PNT2022TMID41014**

MOHAN RAJ P - 612219205005

DEEPA M - 612219205001

JAGATHISHWARAN M - 612219205002

KANMANI PRIYA A - 612219205004

YOKESH P -612219205006

TABLE OF CONTENTS

6. PROJECT PLANNING & SCHEDULING
    - Sprint Planning & Estimation
    - Sprint Delivery Schedule
    - Reports from JIRA
7. CODING & SOLUTIONING (Explain the features added in the project along with code)
    - Feature 1
    - Feature 2
    - Database Schema (if Applicable)
8. TESTING
    - Test Cases
    - User Acceptance Testing
9. RESULTS
    - Performance Metrics
10. ADVANTAGES & DISADVANTAGES
11. CONCLUSION
12. FUTURE SCOPE
13. APPENDX

Source Code

GitHub & Project Demo Link

1.INTRODUCTION

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.

Typically a victim receives a message that appears to have been sent by a known contact or organization. The message contains malicious software targeting the user's computer or has links to direct victims to malicious websites in order to trick them into divulging personal and financial information, such as passwords, account IDs or credit card details.

1.1 Project Overview

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

The importance to safeguard online users from becoming victims of online fraud, divulging confidential information to an attacker among other effective uses of phishing as an attacker's tool, phishing detection tools play a vital role in ensuring a secure online experience for users.

Phishing has a list of negative effects on a business, including loss of money, loss of intellectual property, damage to reputation, and disruption of operational activities. These effects work together to cause loss of company value, sometimes with irreparable repercussions.

2.LITERATURE SURVEY

2.1EXISTING PROBLEM

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.

2.2 REFERENCES

Paper 1 :

Title: Phishing Website Detection Using ML
Year: 2021
Authors: Nikhil K*, Dr. Rajesh D S, Dhanush Raghavan

Description:

In emerging technology, industry, which deeply influence today's security problems, has given a headache to many employers and home users. Occurrences that exploit human vulnerabilities have been on the upsurge in recent years. In these new times there are many security systems being enabled to ensure security is given the outmost priority and prevention to be taken from being hacked by those who are involved in

cyber-offenses and essential prevention is taken as high importance in organization to ensure network security is not being compromised. Cybersecurity employee are currently searching for trustworthy and steadydetection techniques for phishing websites detection. Due to wide usage ofinternet to perform various activities such as online bill payment, bankingtransaction, online shopping, etc. Customer face numerous security threatslike cybercrime. Many cybercrime is being casually executed for examplespam, fraud, identity theft cyber terrorisms and phishing. Among thisphishing is known as the most common cybercrime today. Phishing hasbecome one amongst the top three most current methods of law breakingin line with recent reports, and both frequency of events and user weaknesshas increased in recent years, more combination of all these methodsresult in greater danger of economic damage. Phishing is a socialengineering attack that targets and exploiting the weakness found in thesystem at the user's end. This paper proposes the Agile Unified Process(AUP) to detect duplicate websites that can potentially collect sensitiveinformation about the user. The system checks the blacklisted sites indataset and learns the patterns followed by the phishing websites andapplies it to further given inputs. The system sends a pop-up and an e-mailnotification to the user, if the user clicks on a phishing link and redirects tothe site if it is a safe website. This system does not support real timedetection of phishing sites; user has to supply the website link to thesystem developed with Microsoft Visual Studio 2010 Ultimate and MySQLstocks up data and to implement database in this system.

Paper - 2:

Title: Web Phishing Detection using Machine LearningYear: 2022

Authors: N Kumaran, Purandhar Sri Sai, Lokesh Manikanta

Description:

The current circumstance is that the population's maturity has been wisecracked, causing them to unknowingly give their private information to hackers. Several banned websites have already been established to seem like that of an actual point of contact through obtaining stoners' private information. Passcode, savings account, and shipping information are just a few examples. Late in 2016, the amount of hacking activities was at an all-time high since the company started monitoring this in 2004. The overall identified phishing attacks in 2016 were 1,609. This represents a 65 percent increase over 2015. Within the final quarter of 2004, there would be scamming attempts each month. Machine Learning was used to find the phishing website. The use of machine literacy to surround the supplied features is the basis of Grounded Malware Monitoring Systems. Features are generated by assembling items in a specific order, such as URLs, sphere names, website features, and website content. Because of its nonlinear system, it has a high level of fashion ability in terms of web security, particularly for the detection of anomalies on internet spots. The features retrieved utilizing machine literacy approaches are compared to extracting features through URLs, primary law, or third-party services. A process of machine trust ability on a particularity meant for the reflection of the besieged deceit of stoners through electronic communication is a relevant approach for detecting these attacks.This method can be used to find phishing websites or textbook dispatches sent over email to confuse the victims. This method was

presented by S. Marchal et al. to distinguish Malicious URLs based on the assessment of legitimate point garçon record data. By the off operation or the detection of a malicious site. Open source demonstrates several remarkable characteristics, including high proximity, total autonomy, excellent linguistic flexibility, quickness in choosing, inflexibility towards active phishing, and inflexibility towards development in phishing methods. Mustafa Aydin et al. presented the bracket method to fraudulent site detection that involves rooted websites 'URL properties and evaluating subset- grounded Point selection approaches. For the detection of phishingwebsites, it uses point birth and selecting styles. Fadi Thabtah et al. evaluated vast numbers of ML methods to actual malware datasets and according to many parameters. The goal of this comparison is to highlight the benefits and drawbacks of ML predictive models, as well as their real performance in phishing attempts. Covering approach models are more appropriate as anti-phishing results, according to the experimental results. Muhemmet Baykara et al. developed the Anti Phishing Simulator, which gives data on the phishing discovery challenge as well as how to detect phishing emails. Only utilize the textbook of the e-mail as just a term to execute complicated word processing, according to the study's recommendations.

2.3 PROBLEM STATEMENT DEFINITION

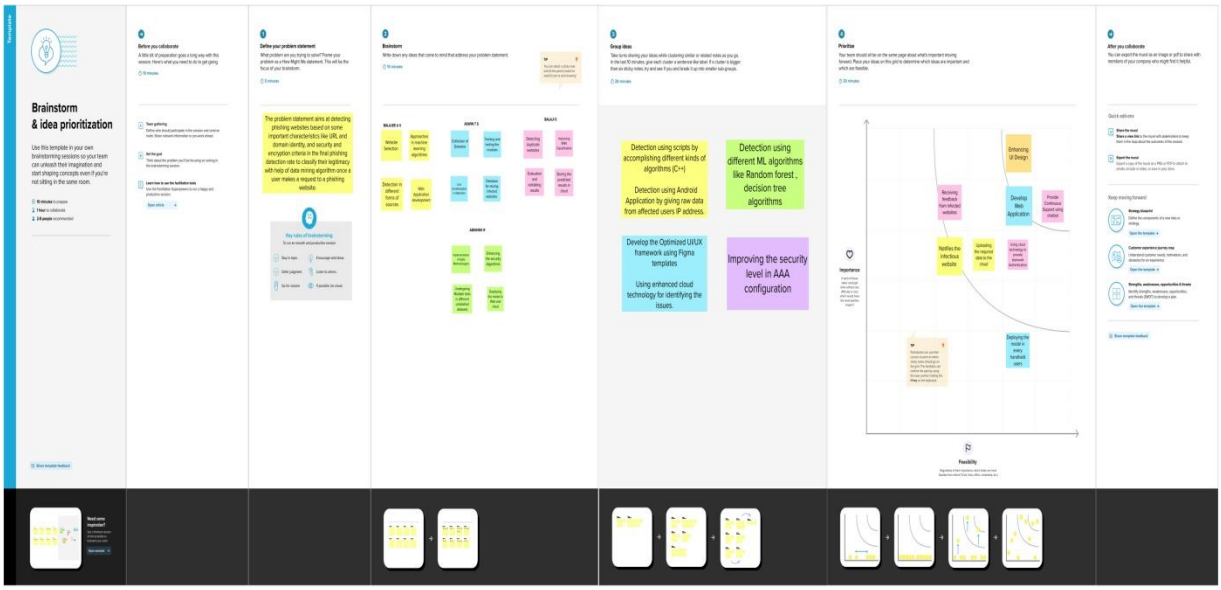
'Phishing sites' are some type of the internet security issues that mainly targets the human vulnerabilities compared to software vulnerabilities. Phishing sites are malicious websites that imitate as legitimate websites or web pages and aim to steal user's personal credentials like user id, password, and financial information. Spotting thesephishing websites is typically a challenging task because phishing is mainly a semantics-based attack, that mainly focus on human vulnerabilities, not the network or software vulnerabilities. Phishing can be elaborated as the process of charming users in order to gain their personal credentials like

user-id’s and Passwords.

3.IDEATION AND PROPOSED SOLUTION

3.1EMPATHY MAP CANVAS

3.2 IDEATION & BRAINSTROMING



3.3 PROPOSED SOLUTION

S.NO.	PARAMETER	DESCRIPTION
1.	Problem Statement (problem to be solved)	A particular challenge in this domain is that notorious hackers are constantly making new strategies to break into our defense measures. The drawback of the existing systems is detecting some minor false positive and false negative results. These disadvantages can be abolished by introducing a much-enhanced feature to feed to the machine learning algorithm that would result in much higher accuracy.
2.	Idea/ Solution Description	We focus on the direct implementation of the project to the chrome extension so that as the user clicks on the particular URL and if that URL is a phishing site then the user gets a pop-up warning message.
3.	Novelty/ Uniqueness	1. Using Machine Learning we developed the Web application as Web Phishing Detection 2. It checks the URL and no of users visited in that particular webpage or website and creates an alert to the user if the website is dangerous or not .
4.	Social Impact/ Customer Satisfaction	1. To spread the cyber awareness on multiple attacks mainly on this phishing attack. 2. An individual can unlearn and relearn this model in various types of aspects in Cyber security and Data theft.
5.	Business Model (Financial Benefit)	1.This model helps Banking and Financial sectors from data loss and data attack which leads to zero financial loss externally. 2.In Business Organization they can use this tool to get rid from cyber attack and can implement how to improve the security when this attack occur next time.
6.	Scalability of Solution	1. We deliver the Good feasible UI/UX design on Web phishing detection. 2. The model is tested and trained in multiple types of datasets to get high accuracy than other algorithms.

3.4 PROBLEM SOLUTION FIT

4.REQUIREMENT ANALYSIS

4.1FUNCTIONAL REQUIREMENTS

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Authentication	Confirmation of Google Firebase
FR-4	User Security	Strong Passwords , 2FA and FIDO2.0 Webaucn
FR-5	User Performance	Usage of Legitimate websites, Optimize Network Traffic

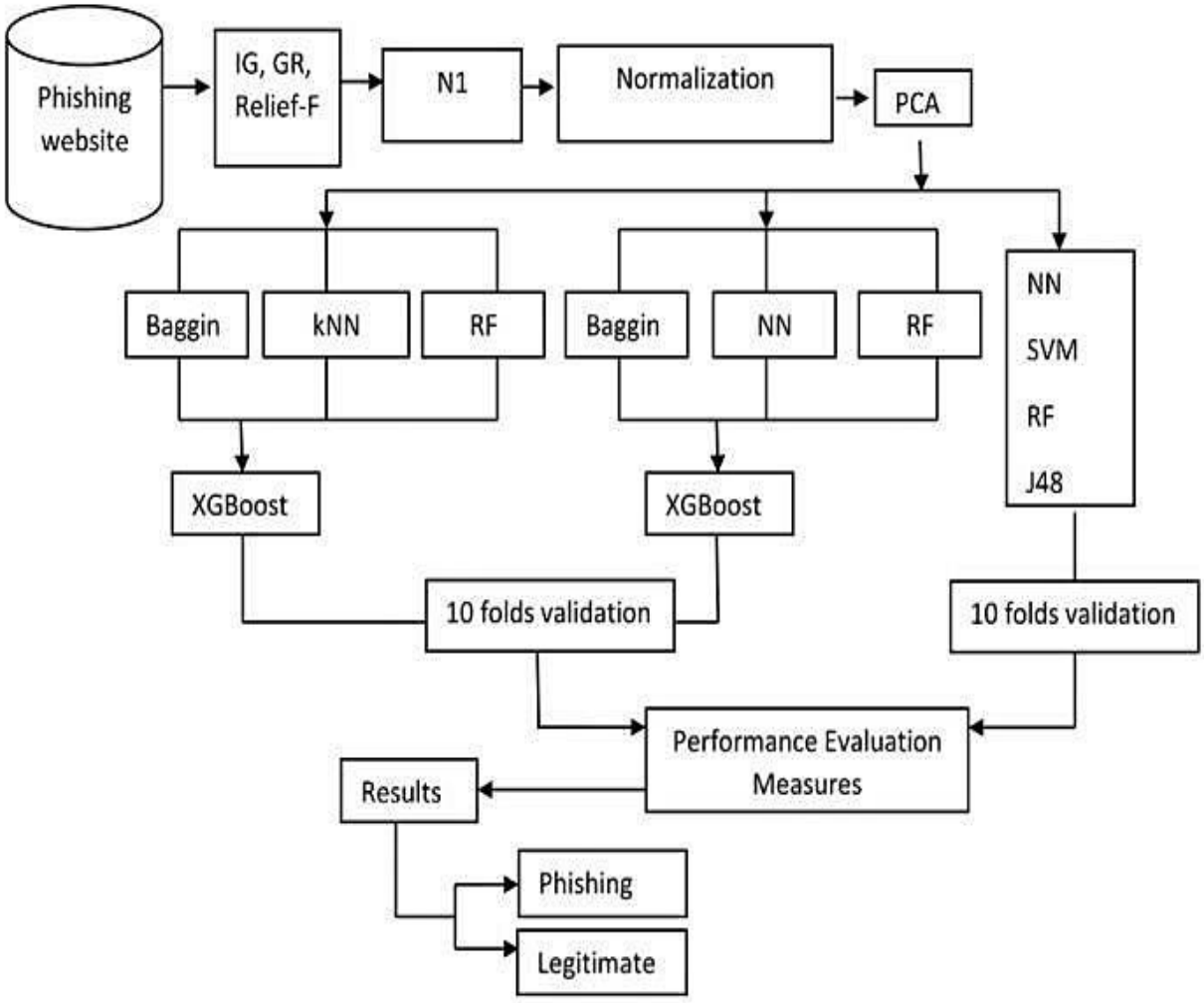
4.2 NON - FUNCTIONAL REQUIREMENTS

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

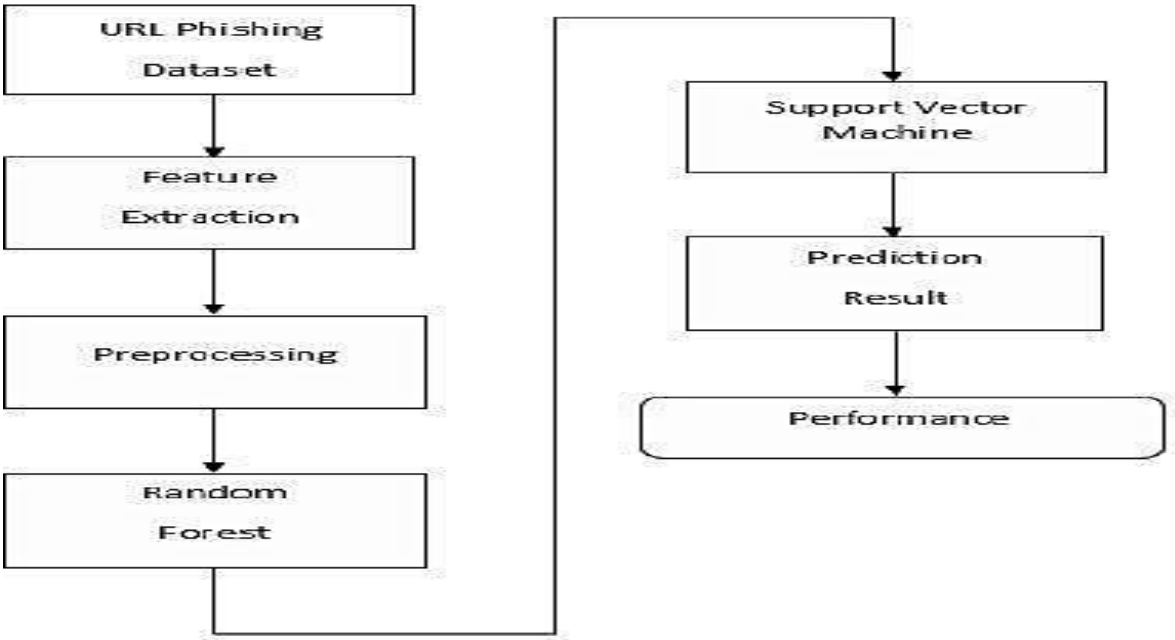
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Responsive UI / UX Design and users can easily configure the settings based on their preference.
NFR-2	**Security**	Implementation of Updated security algorithms and techniques.
NFR-3	**Reliability**	Reliability Factor determines the possibility of a suspected site to be Valid or Fake.
NFR-4	**Performance**	The two main characteristics of a phishing site are that it looks extremely similar to a legitimate site and that it has at least one field to enable users to input their credentials.
NFR-5	**Availability**	It occurs when an attacker, masquerading as a trusted entity, dupes a victim into opening an email, instant message, or text message.
NFR-6	**Scalability**	Scalable detection and isolation of phishing, the main ideas are to move the protection from end users towards the network provider and to employ the novel bad neighbourhood concept, in order to detect and isolate both phishing e mail senders and phishing web servers.

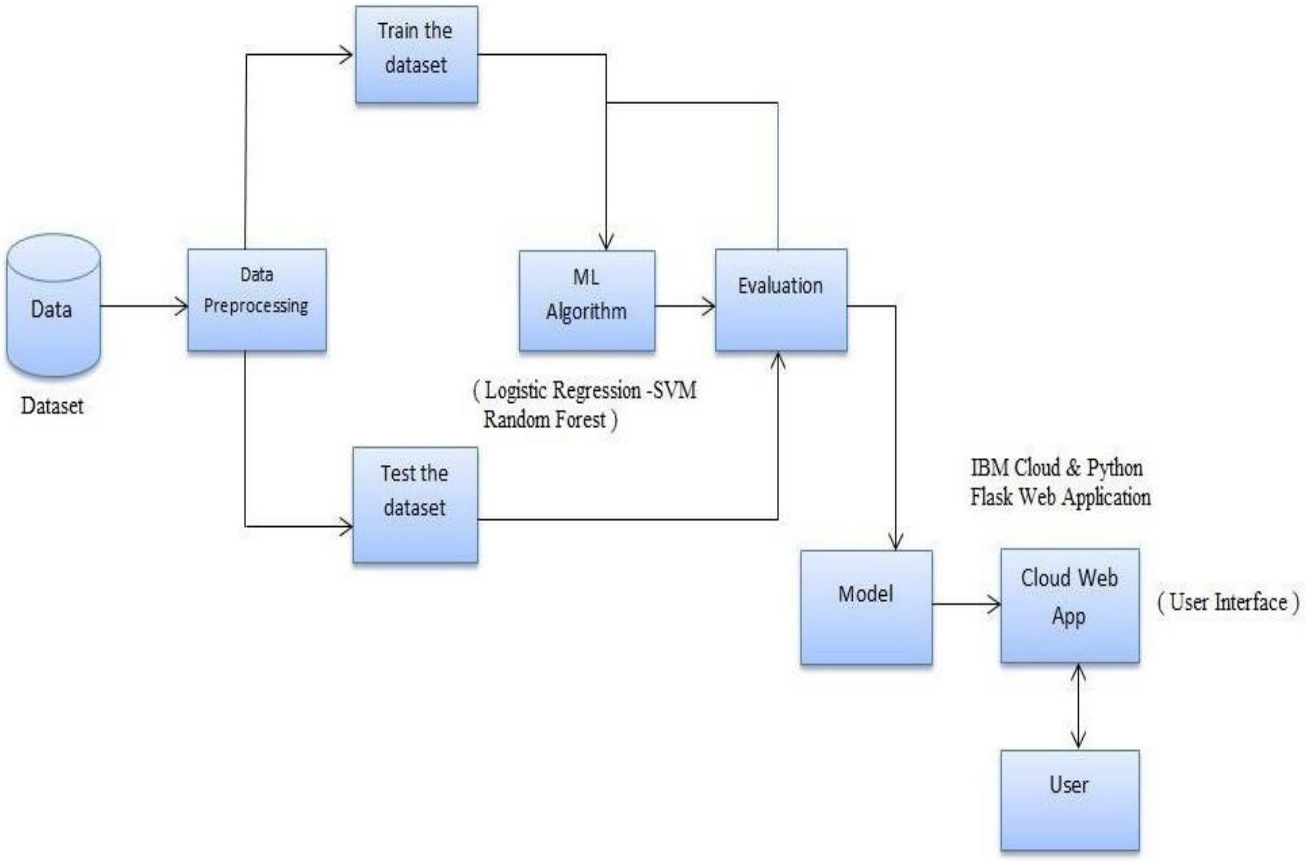
5.PROJECT DESIGN

5.1 DATA FLOW DIAGRAM



5.2 SOLUTION AND TEECHNICAL ARCHITECTURE





5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register my personal details only in official websites.	I can access my account / dashboard	Medium	Sprint-1
		USN-2	As a user, I should create strong passwords.	I can access my account securely	High	Sprint-1
		USN-3	As a user, I can register in websites which doesn't navigate me to any other websites.	I can store the data in legitimate website	Low	Sprint-2
	Login	USN-4	As a user, I can login into required websites.	I can access my account	Low	Sprint-1
Customer (Mobile user)	Registration	USN-5	As a user, I can register with verification code.	Authorized Login	High	Sprint-1
		USN-6	As a user, I should not register at unknown or random calls.	I can be prevented from Cyber Attacks	Medium	Sprint-1
		USN-7	As a user, I should not register in other devices.	I can access in my authorized device.	Low	Sprint-2
Administrator		USN-8	Admin should maintain his/her database securely.	Prevented from Phishing Attacks	High	Sprint-2
Customer Care		USN-9	As a user, If my account is Phished or Attacked.	I can report / Complain	High	Sprint-1
		USN-10	As a user, I should not take others information	I can be punished for it.	Medium	Sprint-1

6.PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND EXECUTION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	User input	USN-1	User inputs an URL in the required field to check its validation.	1	Medium	Deepa.M
Sprint-1	Website Comparison	USN-2	Model compares the websites using Blacklist and Whitelist approach.	1	High	Yokesh.P
Sprint-2	Feature Extraction	USN-3	After comparison, if none found on comparison then it extract feature using heuristic and visual similarity.	2	High	Jagathishwaran.M
Sprint-2	Prediction	USN-4	Model predicts the URL using Machine learning algorithms such as logistic Regression, KNN.	1	Medium	Mohan Raj.P
Sprint-3	Classifier	USN-5	Model sends all the output to the classifier and produces the final result.	1	Medium	Kanmani Priya.A
Sprint-4	Announcement	USN-6	Model then displays whether the website is legal site or a phishing site.	1	High	Jagathishwaran.M
Sprint-4	Events	USN-7	This model needs the capability of retrieving and displaying accurate result for a website.	1	High	Mohan Raj.P

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	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 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

7.CODING & SOLUTIONING

```
#importing required libraries

import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle
import requests
import inputScript



# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = ""
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}
```

```
#load model
app = Flask(__name__)
model = pickle.load(open("model.pkl", 'rb'))

#Redirects to the page to give the user input URL.
@app.route('/')
def predict():
    return render_template('index.html',result="")

#Fetches the URL given by the URL and passes to inputScript
@app.route('/',methods=['POST'])
def y_predict():
    '''
    For rendering results on HTML GUI
    '''
    url = request.form['url']
    checkprediction = inputScript.main(url)
    print(url)
    print(checkprediction)
    prediction = model.predict(X=checkprediction)
    requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments//predictions?version=2022-11-06', json=prediction,
        headers={'Authorization': 'Bearer ' + mltoken})
    print(prediction)
    output=prediction[0]
    print(output)
    if(output==1):
        pred="Your are safe!!  This is a Legitimate Website."
```

8.TESTING

8.1 TEST CASES

				Date	15-Nov-22								
				Team ID	PNT2022TMID41014								
				Project Name	Project - Web Phishing Detection								
				Maximum Marks	4 marks								
Test case ID	**Feature Type**	**Component**	**Test Scenario**	**Pre-Requisite**	**Steps To Execute**	**Test Data**	**Expected Result**	**Actual Result**	**Status**	**Comments**	**TC for Automation(Y/N)**	**BUG ID**	**Executed By**
LoginPage_TC_OO1	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		Mohan Raj.P
LoginPage_TC_OO2	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		Jagathishwaran.M
LoginPage_TC_OO3	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		Deepa.M
LoginPage_TC_OO4	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		Kanmani Priya.A
LoginPage_TC_OO5	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://avbalajee.github.io/welcome 2. totalpad.com 3. https://www.klnce.edu 4. salescript.info 5. https://www.google.com/ 6. delgets.com	User can able to identify the websites whether it is secure or not	Working as expected	Pass		N		Yokesh.P

8.2 USER ACCEPTANCE TESTING

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [Web Phishing Detection] project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

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

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

3. Test Case Analysis

This report shows the number of 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

9.RESULTS

9.1PERFORMANCE METRICS

Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot
1.	Metrics	**Classification Model:** **Gradient Boosting Classification** Accuray Score- 97.4%	
2.	Tune the Model	Hyperparameter Tuning - 97% Validation Method – KFOLD & Cross Validation Method	

1. METRICS:
CLASSIFICATION REPORT:

```
In [52]: #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
```

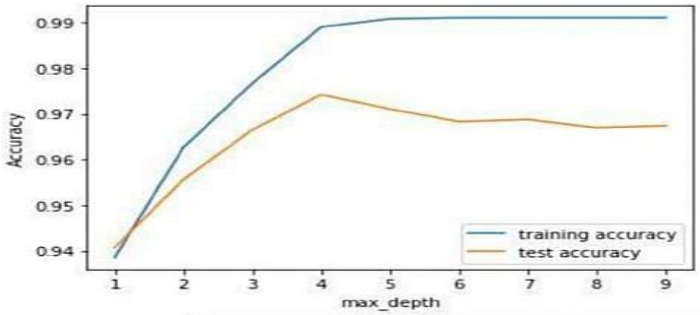
2. TUNE THE MODEL – HYPERPARAMETER TUNING

```
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_))
```

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. TUNE THE MODEL – HYPERPARAMETER 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 & Cross Folding

Wilcoxon signed-rank test

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.ADVANTANGES & DISADVANTAGES

ADVANTAGES

- It takes the Load off the Security team
- Improve on Inefficiencies of SEG and Phishing Awareness Training
- Password Management made Easy

DISADVANTAGES

- Low Detection Accuracy
- False Alarm

11.CONCLUSION

This paper presented an intelligent phishing detection and protection scheme by employing a new approach using the integrated features ofimages, frames and text of phishing websites. An efficient ANFIS algorithm was developed, tested and verified for phishing website detection and protection based on the schemes proposed in Aburrous et al. (2010) and Barraclough and Sexton (2015). A set of experiments was performed using 13,000 available datasets. The approach showed an accuracy of 98.3%, which so far, is the best-integrated solutions for web-phishing detection and protection.

The primary contribution of this study is the integration of hybrid features that have been extracted from text, images and frames and that are then used to develop a robust ANFIS solution. Future work will include using another algorithm like deep-learning for phishing web page detection and compare the effectiveness with the current result. More also, a web browser plug-in will be developed based on an efficient algorithm to detect phishing website and thus protect users in real time.

12.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 toget maximum accuracy. We also plan to explore various 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 improve the performance of the system. In particular, we extract features from URLs and pass it through the various classifiers.

We have seen that existing system gives less accuracy so we proposed a new phishing method that employs URL based features and also we generated classifiers through several machine learning Future work will consist in releasing components of the tools as an add-on for a Web browser such as Mozilla Firefox. In addition, the technique proposed, which is complementary to that introduced in thispaper, will be merged to create a phishing detection system with a larger scope of action. We also plan to release the analytics related part in a larger Big Data security analytics stack, which is under current development in our lab.

13.APPENDIX

SOURCE CODE

MODEL CREATION

```
import regex
from tldextract import extract
import socket
from bs4 import BeautifulSoup
import urllib.request
import whois
import requests
import favicon
import re
from googlesearch import search



#checking if URL contains any IP address. Returns -1 if contains else returns 1
def having_IPhaving_IP_Address(url):
    match=regex.search(
  '(([01]?\\d\\d?|2[0-4]\\d|25[0-5])\\.([01]?\\d\\d?|2[0-4]\\d|25[0-5])\\.([01]?\\d\\d?|2[0-4]\\d|25[0-5])\\.([01]?\\d\\d?|2[0-4]\\d|25[0-5])\\/)|'  #IPv4
                '((0x[0-9a-fA-F]{1,2})\\.(0x[0-9a-fA-F]{1,2})\\.(0x[0-9a-fA-F]{1,2})\\.(0x[0-9a-fA-F]{1,2})\\/)'  #IPv4 in hexadecimal
                '(?:[a-fA-F0-9]{1,4}:){7}[a-fA-F0-9]{1,4}',url)     #Ipv6
    if match:
        #print match.group()
        return -1
    else:
        #print 'No matching pattern found'
        return 1
```

```
def URLURL_Length (url):
    length=len(url)
    if(length<=75):
        if(length<54):
            return 1
        else:
            return 0
    else:
        return -1


#Checking with the shortening URLs.
#Returns -1 if any shortening URLs used.
#Else returns 1
def Shortining_Service (url):
    match=regex.search('bit\.ly|goo\.gl|shorte\.st|go2l\.ink|x\.co|ow\.ly|t\.co|tinyurl|tr\.im|is\.gd|cli\.gs|'
                       'yfrog\.com|migre\.me|ff\.im|tiny\.cc|url4\.eu|twit\.ac|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|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',url)
    if match:
        return -1
    else:
        return 1
```

FLASK APP

```
#importing required libraries

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

import inputScript

#load model
app = Flask(__name__)
model = pickle.load(open("model.pkl", 'rb'))

#Redirects to the page to give the user input URL.
@app.route('/')
def predict():
    return render_template('index.html',result="")

#Fetches the URL given by the URL and passes to inputScript
@app.route('/',methods=['POST'])
def y_predict():
    '''
    For rendering results on HTML GUI
    '''
    url = request.form['url']
    checkprediction = inputScript.main(url)
    print(url)
    print(checkprediction)
```

HOME PAGE(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">

    <!-- 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>Web Phising Detection</title>
</head>

<body class="bg-dark">
<div class="container mt-5">
    <div>
        <center>
        <div class="form col-md text-light" id="form1">
            <center>
            <h2>Is the URL safe to open?</h2>
            <br>
            <form action="/" method ="post" autocomplete="off">
                <input type="text" class="form-control w-50" name ='url' id="url" placeholder="Enter URL" required="" />
                <br>
                <button class="btn btn-info mt-2" role="button" >Check here</button>
            </form>
        </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 defer>
        document.querySelector("#btn1").style.display = "none";
        let result = '{{result}}';
        if(result!==undefined || result!==null){
            console.log(result)
            document.getElementById("prediction").innerHTML = result;
            document.getElementById("btn1").style.display="inline-block";
        }
    </script>

</body>
</html>
```

GITHUB

https://github.com/IBM-EPBL/IBM-Project-33705-1660225682

PROJECT DEMO

https://drive.google.com/file/d/1ykKEuYNl2G--RfkgLOMYPPerDvjtBHP5/view?usp=share_link