AVS ENGINEERING COLLEGE

We Create Responsible Engineers

ISO 9001:2015 Certified Institution

Military Road, Ammapet, Salem - 636 003.

WEB PHISHING DETECTION

TEAM ID : PNT2022TMID30885

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

<u>PRESENTED BY</u>

1.KIRUTHIKA M	-	Team Leader
2.ARUNADEVI A	-	Team Member 1
3.GOWSHIKA N	-	Team Member 2
4.JEEVA B	-	Team Member 3
5.MADHUMITHA N	-	Team Member 4

ABSTRACT :

A web service is one of the most important Internet communications software services. Using fraudulent methods to get personal information is becoming increasingly widespread these days. However, it makes our lives easier, it leads to numerous security vulnerabilities to the Internet's private structure. Web phishing is just one of the many security risks that web services face. Phishing assaults are usually detected by experienced users however, security is a primary concern for system users who are unaware of such situations. Phishing is the act of portraying malicious web runners as genuine web runners to obtain sensitive information from the end-user. Phishing is currently regarded as one of the most dangerous threats to web security. Vicious Web sites significantly encourage Internet criminal activity and inhibit the growth of Web services. As a result, there has been a tremendous push to build a comprehensive solution to prevent users from accessing such websites. We suggest a literacy-based strategy to categorize Web sites into three categories: benign, spam, and malicious. Our technology merely examines the Uniform Resource Locator (URL) itself, not the content of Web pages. As a result, it removes run-time stillness and the risk of drug users being exposed to cyber surfer-based vulnerabilities.

1.INTRODUCTION:

1.1 PROJECT OVERVIEW:

We do the majority of our work during the day accomplish sales and activities in industries such as business, medical, academia, information, finance, aeronautics, exploration, infrastructure, enjoyment, and welfare programs promptly. With the network can now be readily accessible to the web any day at any time. on internet platforms. Using a system and the broadband connection in a variety of ways makes our work and personal lives easier. It provides us to Although this arrangement is extremely convenient, it has highlighted major information security bugs. As a result, now need to drug addicts in cyber to take precautions from computer security is revealed.

Cybercriminals, rovers, or non-malicious (fair-limited)bushwhackers and data theft are all capable of carrying out attacks.

1.2 PURPOSE:

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.

2.LITERATURE SURVEY:

2.1 EXISTING PROBLEM:

Machine learning methods were imported using the Scikit-learn library. Each classification is performed using a training set, and the performance of the classifiers is evaluated using a testing set. The accuracy score of classifiers was calculated to assess their performance.

2.2 REFERENCES

1. **Title:** Detection of Phishing Websites by Using Machine Learning-Based URL Analysis.

Author: Mehmet Korkmaz, Ozgur KoraySahingoz, BanuDiri.

Year: 2020

Techniques Used: XGBOOST,RF , LR ,KNN,SVM,DTANN,NB

Description: A machine learning-based phishing detection system by using eight different algorithms to analyze the URLs, and three different datasets to compare the results with other works. The experimental results depict that the proposed models have an outstanding performance with a success rate.

2. **Title:** A Deep Learning-Based Framework for Phishing Website Detection

Author: Lizhen Tang , Qusay H. Mahmoud

Year: 2021

Techniques Used: RNN-GRU, web browser extension.

Description: The author briefed that they have implemented the framework as a browser plugin capable of determining whether there is a phishing risk in real-time when the user visits a web page and gives a warning message. It combines multiple strategies to improve accuracy, reduce false alarm rates, and reduce calculation time, including whitelist filtering, blacklist interception, and machine learning (ML) prediction.

3. **Title:** Detection of Phishing Websites from URLs by using Classification Techniques on WEKA

Author: BuketGeyik, Kubra Erensoy, EmreKocyigit

Year: 2021

Techniques Used: Machine learning, classification algorithms, phishing detection, cybersecurity

Description: The anti-phishing method has been developed by detecting the attacks made with the technologies used. we combined the websites used by phishing attacks into a dataset, then we obtained some results using 4 classification algorithms with this dataset.

4. **Title:** Real Time Detection of Phishing Websites Author: Abdulghani Ali Ahmed, Nurul Amirah Abdullah Year: 2016

Techniques Used: URL,Yahoo Datasets ,Phishing Detection

Description: A detection technique of phishing websites based on checking Uniform Resources Locators (URLs) of web pages. The proposed solution is able to distinguish between the legitimate web page and fake web page by checking the Uniform Resources Locators (URLs) of suspected web pages. URLs are inspected based on particular characteristics to check the phishing web pages. The detected attacks are reported for prevention. The performance of the proposed solution is evaluated using Phish tank and Yahoo directory datasets.

5. **Title:** Phishing URL Detection: A Real-Case Scenario Through Login URLs

Author: Manuel sánchez-paniagua , eduardo fidalgo fernández ,enrique alegre ,wesam alnabki, víctor gonzález-castro

Year: 2022

Techniques Used: Machine learning and deep learning approaches , cybercrime , phishing detection, url.

Description: The list provided on that website only contains the domain names, extracted the complete URL. To reach the login page from a website, It used the Selenium web driver and Python, checking buttons or links that could lead to the login form web page. Once we found the presumptive login and inspected if the form had a password field in order to confirm whether it was a login form. Otherwise, it was not added to the dataset. In this, collected reported phishing URLs from Phishtank.

6. **Title:** A Novel Machine Learning Approach to Detect Phishing Websites

Author: Ishant Tyagi, Jatin Shad, Shubham Sharma, Siddharth Gaur, Gagandeep Kaur

Year: 2018

Techniques Used: Decision Tree, Random Forest, Gradient Boosting , Generalized Linear Model, prediction for a new URL.

Description: In this technique ,they determined most targeted brand names and their legit URL via Google and their real phishing URLs from PhishTank website. Those extracted using python and used for prediction for a new URL. Input URL, Extract 30 features of URL, Use these features for predictive analysis,It checks whether it obtains positive or negative output.if negative it notifies the user that the website is phishing otherwise Notify the user that the website is safe.

2.3 PROBLEM STATEMENT DEFINITION:

Phishing is a major problem, which uses both social engineering and technical deception to get users' important information such as financial data, emails, and other private information. Phishing exploits human vulnerabilities; therefore, most protection protocols cannot prevent the whole phishing attacks.

3. IDEATION AND PROPOSED SOLUTION:

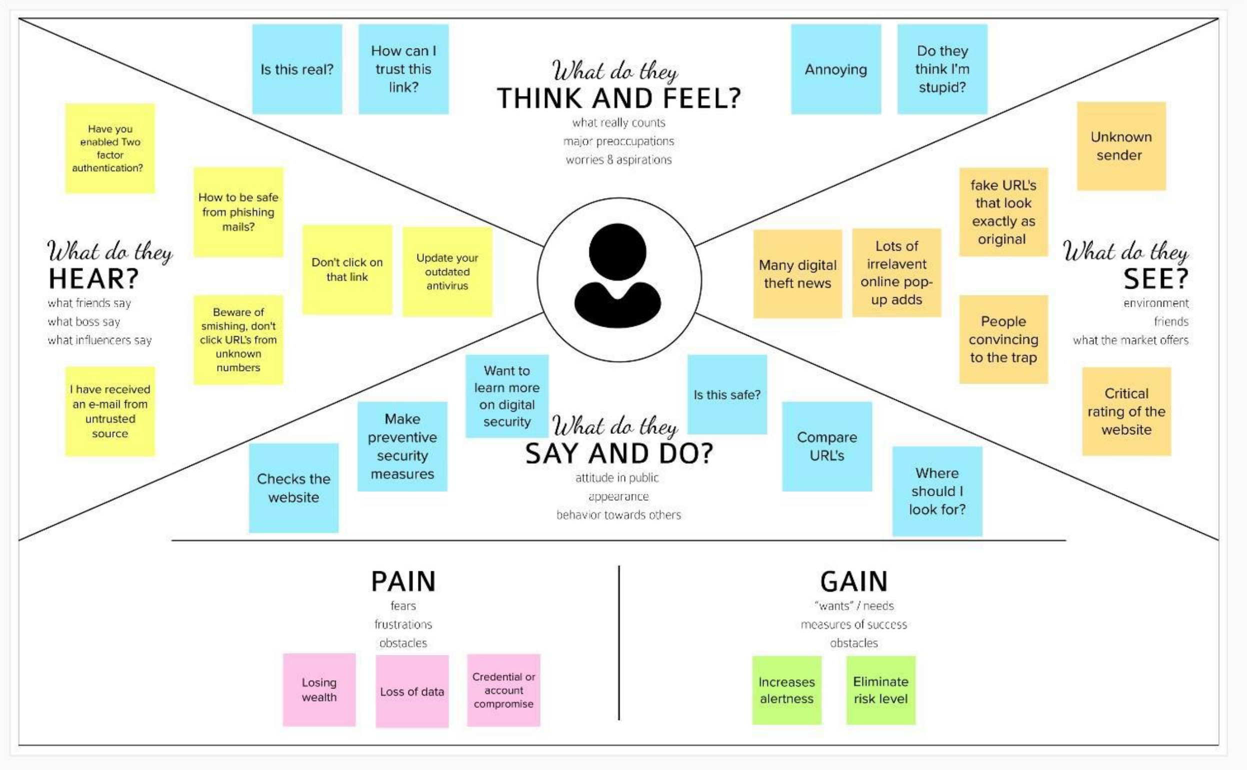
3.1 EMPATHY MAP CANVAS:

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user’s

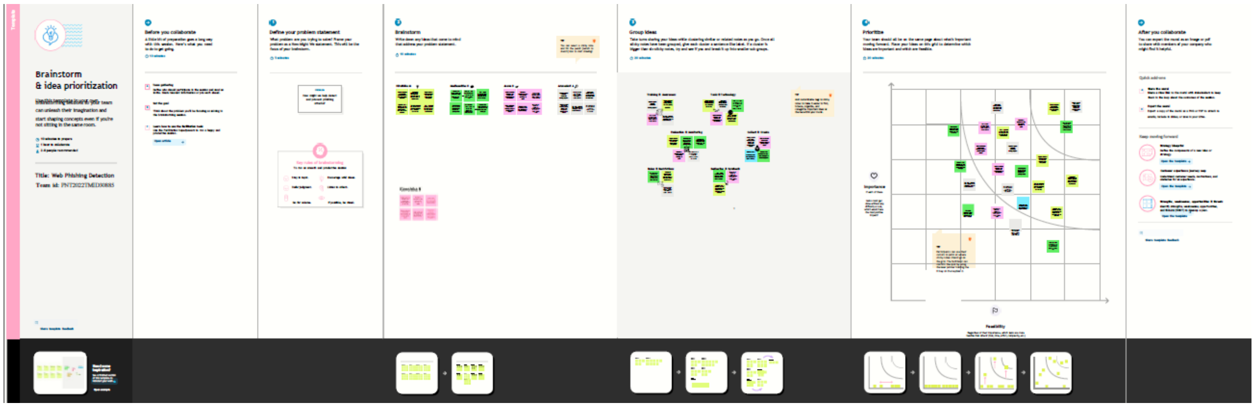
behaviours and attitudes.

It is a useful tool to helps teams better understand their users.

Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user’s perspective along with his or her goals and challenges.



3.2 IDEATION & BRAINSTORMING



Brainstorm
& idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

Title: Web Phishing Detection
Team id: PNT2022TMID30885

3.3 PROPOSED SOLUTION:

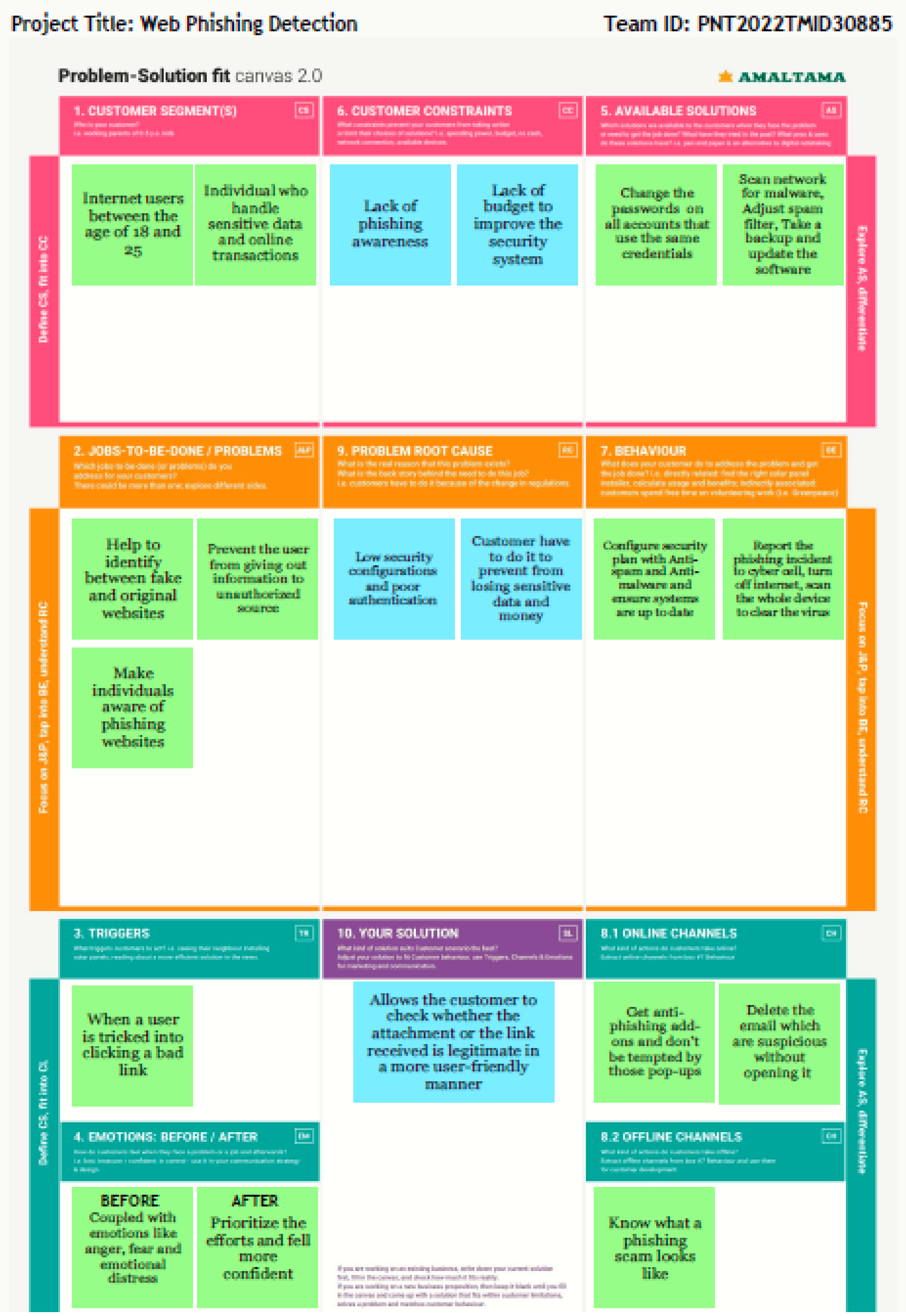
Proposed Solution Template:

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

S.No	Parameter	Description
1	Problem Statement (Problem to be solved)	The Phish report states that around 74% people were sent fraudulent messages every month. While this cannot be stopped completely, some preventable actions can be taken. To prevent and predict phishing websites, we proposed an intelligent, flexible, and effective system that is based on using classification Data mining algorithm.
2	Idea / Solution description	In a replicated website there must have some flaws, The 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.
3	Novelty / Uniqueness	In this techy world, there are many technologies offer solution to protect ourselves from phishing attacks, But the data mining algorithm used in this system provides better performance as compared to other traditional classification algorithms.
4	Social Impact / Customer Satisfaction	The proposes help the user to safely make online transaction without any fear of losing money or sensitive data to the attacker and help them gain some awareness of cyber- threat.
5	Business Model (Revenue Model)	The number of visitors to the website becomes the number of opportunities the business has at giving an impression, generating qualified

		leads, sharing the brand, and building relationship.
6	Scalability of the Solution	The features can progressively increase to scan the attachment, file hash, IP address, etc.,

3.4 PROBLEM SOLUTION FIT:



Project Title: Web Phishing Detection

Team ID: PNT2022TMID30885

Problem-Solution fit canvas 2.0

1. CUSTOMER SEGMENT(S) CS

- Internet users between the age of 18 and 25
- Individual who handle sensitive data and online transactions

6. CUSTOMER CONSTRAINTS CC

- Lack of phishing awareness
- Lack of budget to improve the security system

5. AVAILABLE SOLUTIONS AS

- Change the passwords on all accounts that use the same credentials
- Scan network for malware, Adjust spam filter, Take a backup and update the software

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.

- Help to identify between fake and original websites
- Prevent the user from giving out information to unauthorized source
- Make individuals aware of phishing websites

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists? What is the back story behind the need to do this job? i.e. customers have to do it because of the change in regulations.

- Low security configurations and poor authentication
- Customer have to do it to prevent from losing sensitive data and money

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done? i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

- Configure security plan with Anti-spam and Anti-malware and ensure systems are up to date
- Report the phishing incident to cyber cell, turn off internet, scan the whole device to clear the virus

3. TRIGGERS TR

- When a user is tricked into clicking a bad link

4. EMOTIONS: BEFORE / AFTER EM

- BEFORE
Coupled with emotions like anger, fear and emotional distress
- AFTER
Prioritize the efforts and fell more confident

10. YOUR SOLUTION SL

- Allows the customer to check whether the attachment or the link received is legitimate in a more user-friendly manner

8.1 ONLINE CHANNELS CH

- Get anti-phishing add-ons and don't be tempted by those pop-ups
- Delete the email which are suspicious without opening it

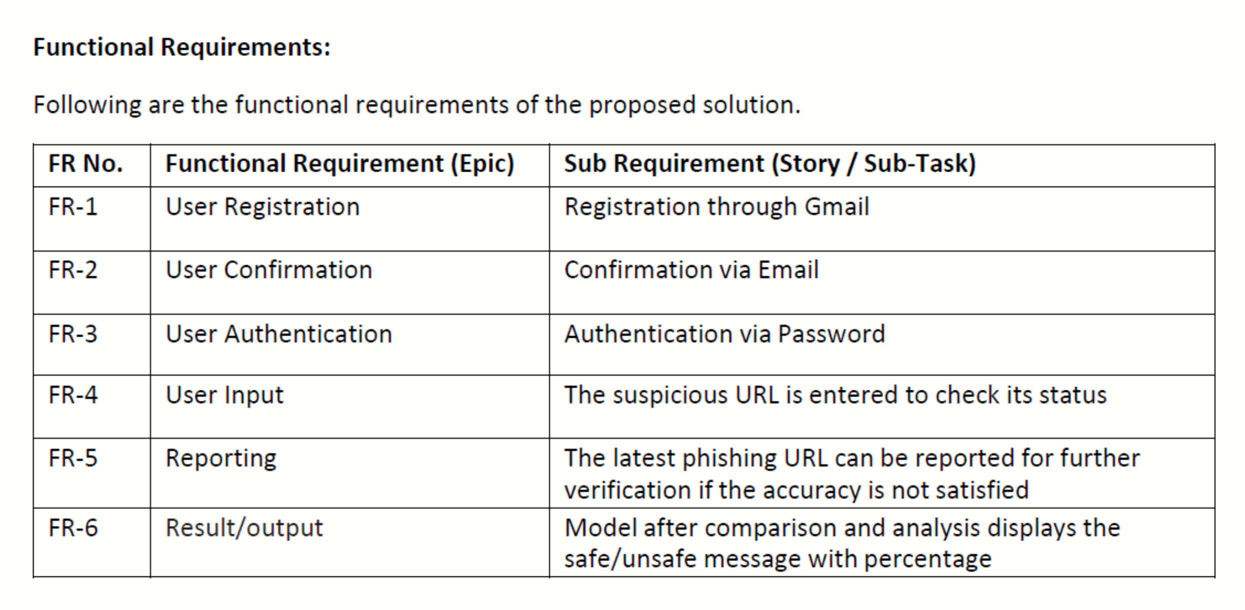
8.2 OFFLINE CHANNELS CH

- Know what a phishing scam looks like

Define CS, fit into CC
Explore AS, differentiate
Focus on J&P, tap into BE, understand RC
Focus on J&P, tap into BE, understand RC
Define CS, fit into CL
Explore AS, differentiate

4. REQUIREMENT ANALYSIS:

4.1 FUNCTIONAL REQUIREMENTS:

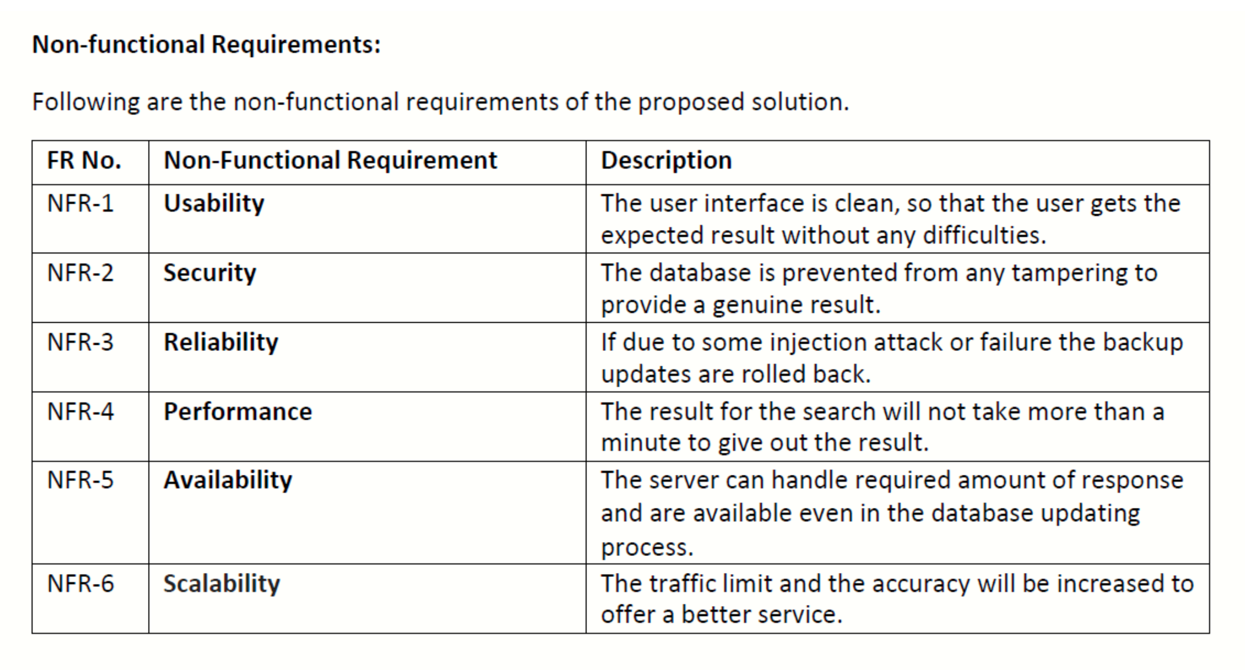


Functional Requirements:

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 Gmail
FR-2	User Confirmation	Confirmation via Email
FR-3	User Authentication	Authentication via Password
FR-4	User Input	The suspicious URL is entered to check its status
FR-5	Reporting	The latest phishing URL can be reported for further verification if the accuracy is not satisfied
FR-6	Result/output	Model after comparison and analysis displays the safe/unsafe message with percentage

4.2 NON FUNCTIONAL REQUIREMENTS:



Non-functional Requirements:

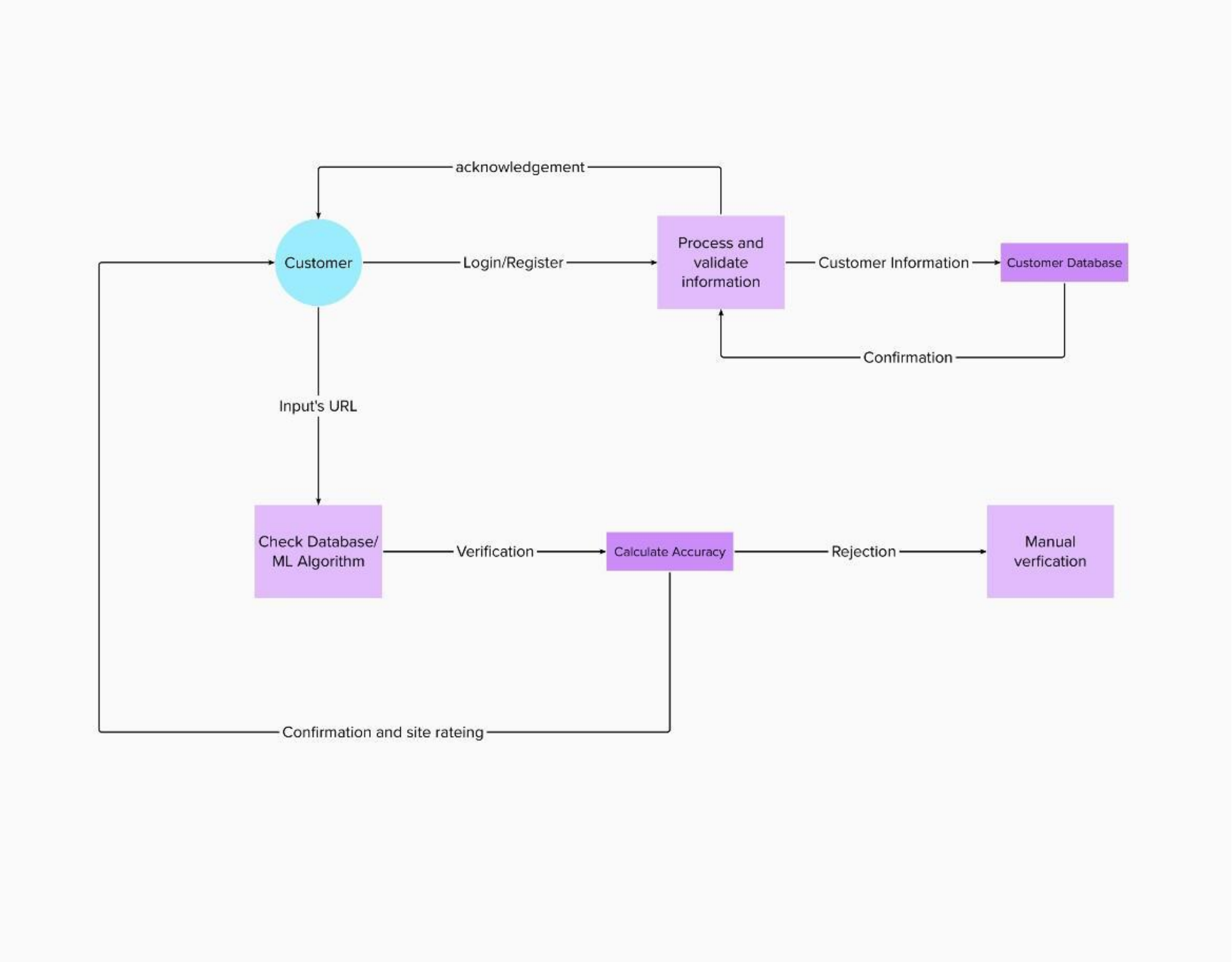
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The user interface is clean, so that the user gets the expected result without any difficulties.
NFR-2	**Security**	The database is prevented from any tampering to provide a genuine result.
NFR-3	**Reliability**	If due to some injection attack or failure the backup updates are rolled back.
NFR-4	**Performance**	The result for the search will not take more than a minute to give out the result.
NFR-5	**Availability**	The server can handle required amount of response and are available even in the database updating process.
NFR-6	**Scalability**	The traffic limit and the accuracy will be increased to offer a better service.

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.

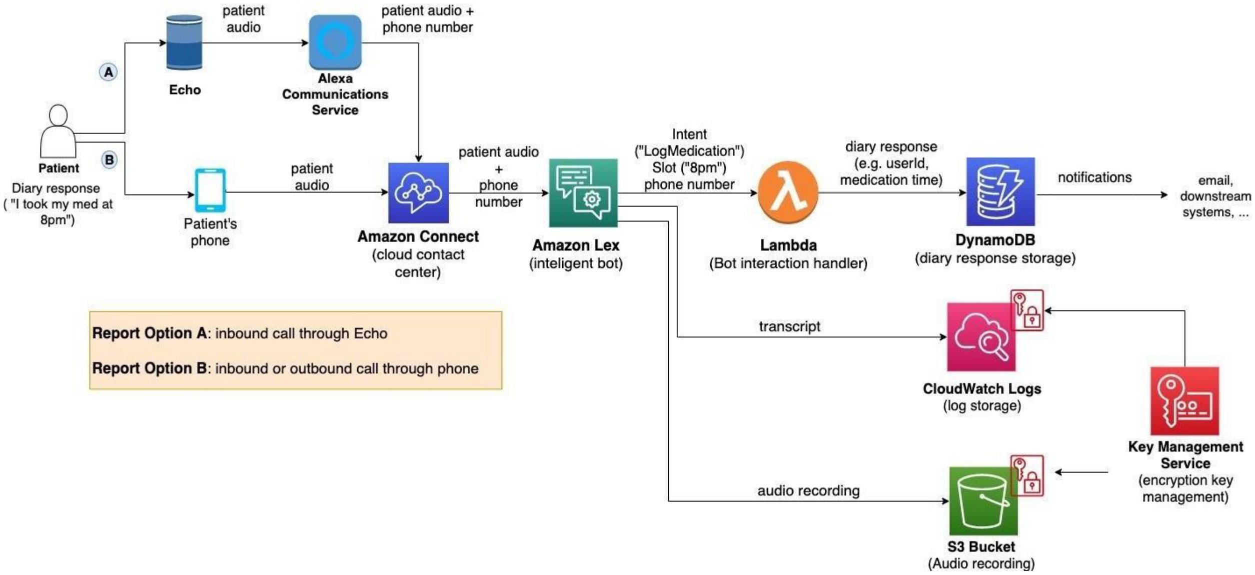


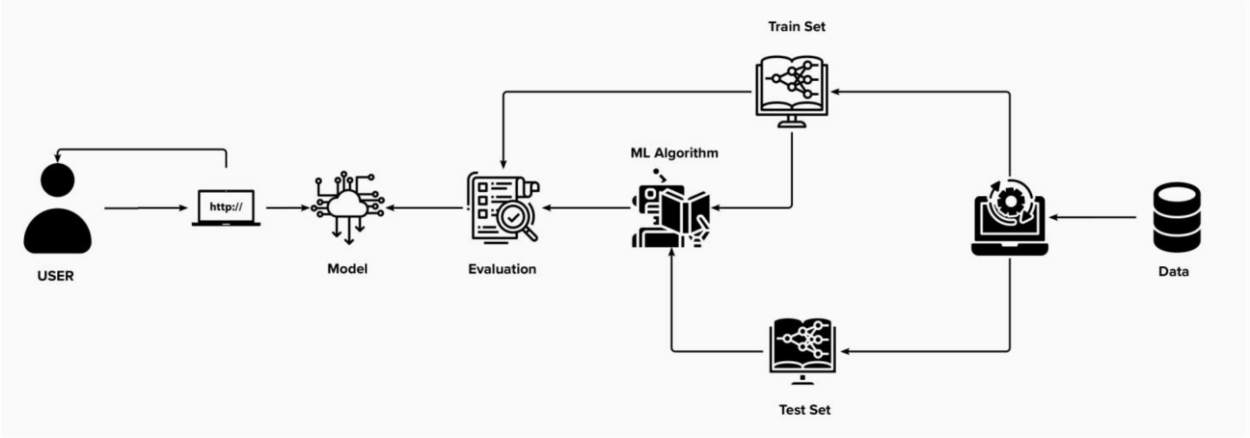
5.2 SOLUTIONS AND TECHNICAL ARCHITECTURE:

SOLUTION ARCHITECTURE

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.


patient audio
patient audio + phone number
Echo
Alexa Communications Service
A
B
Patient
Diary response ("I took my med at 8pm")
patient audio
Patient's phone
patient audio + phone number
Amazon Connect (cloud contact center)
Intent ("LogMedication") Slot ("8pm") phone number
Amazon Lex (inteligent bot)
Lambda (Bot interaction handler)
diary response (e.g. userId, medication time)
DynamoDB (diary response storage)
notifications
email, downstream systems, ...
transcript
CloudWatch Logs (log storage)
Key Management Service (encryption key management)
audio recording
S3 Bucket (Audio recording)
Report Option A: inbound call through Echo
Report Option B: inbound or outbound call through phone



Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

Example: Order processing during pandemics for offline mode

Reference: https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/

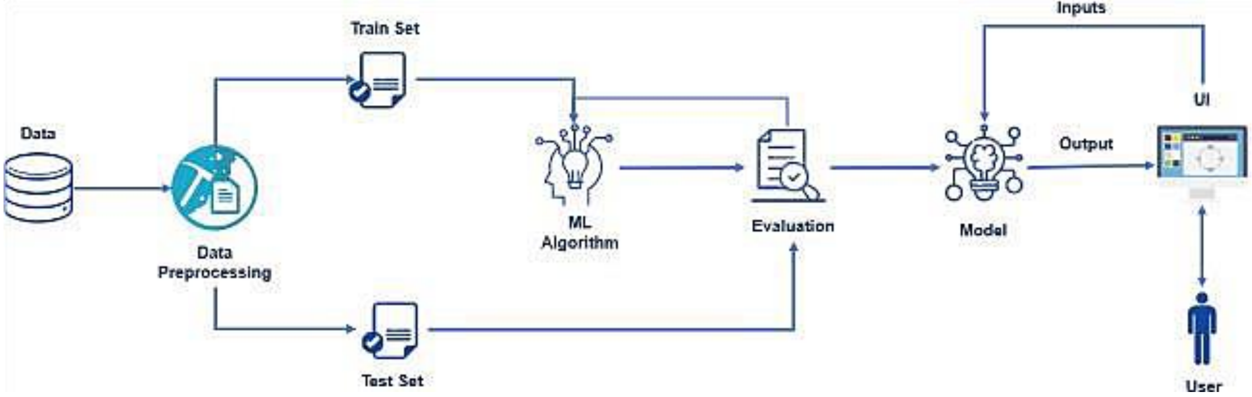
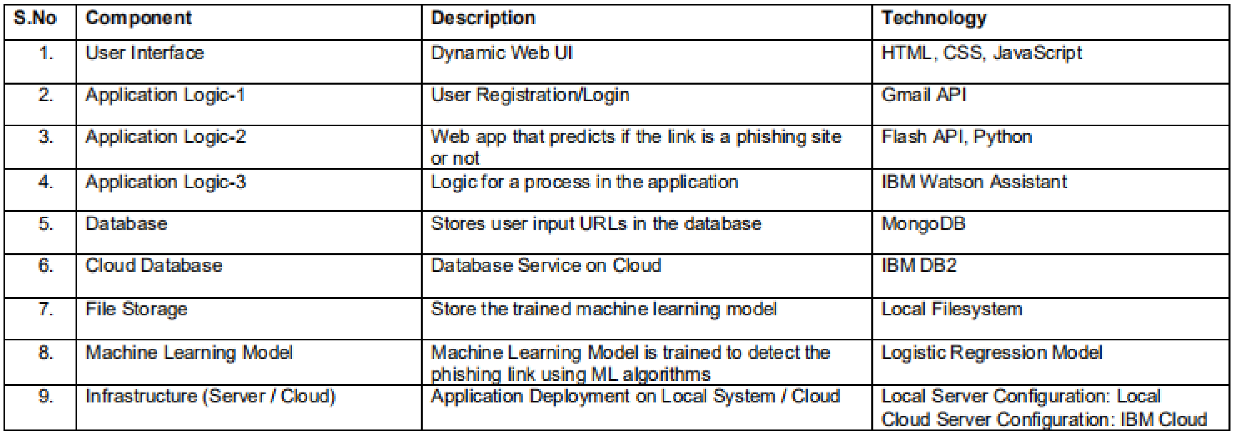
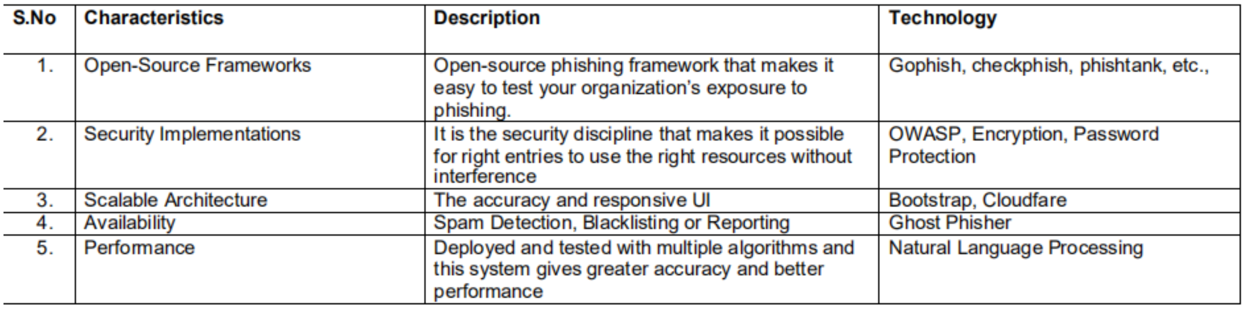


Table-1 : Components & Technologies:



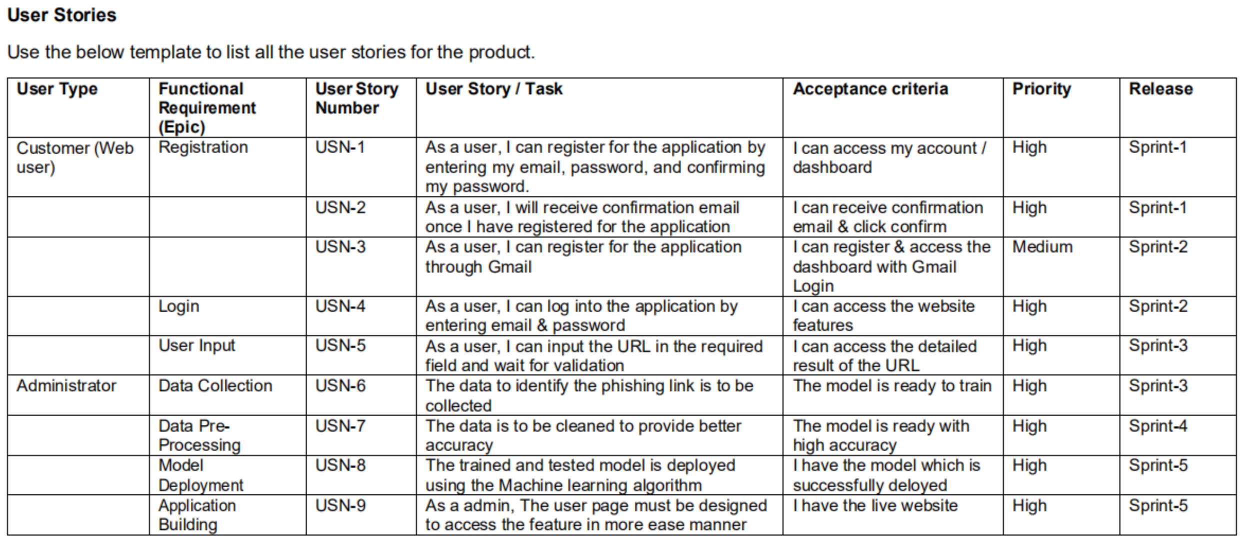
S.No	Component	Description	Technology
1.	User Interface	Dynamic Web UI	HTML, CSS, JavaScript
2.	Application Logic-1	User Registration/Login	Gmail API
3.	Application Logic-2	Web app that predicts if the link is a phishing site or not	Flash API, Python
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Database	Stores user input URLs in the database	MongoDB
6.	Cloud Database	Database Service on Cloud	IBM DB2
7.	File Storage	Store the trained machine learning model	Local Filesystem
8.	Machine Learning Model	Machine Learning Model is trained to detect the phishing link using ML algorithms	Logistic Regression Model
9.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud	Local Server Configuration: Local Cloud Server Configuration: IBM Cloud

Table-2: Application Characteristics:



S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Open-source phishing framework that makes it easy to test your organization's exposure to phishing.	Gophish, checkphish, phishtank, etc.,
2.	Security Implementations	It is the security discipline that makes it possible for right entries to use the right resources without interference	OWASP, Encryption, Password Protection
3.	Scalable Architecture	The accuracy and responsive UI	Bootstrap, Cloudfare
4.	Availability	Spam Detection, Blacklisting or Reporting	Ghost Phisher
5.	Performance	Deployed and tested with multiple algorithms and this system gives greater accuracy and better performance	Natural Language Processing

5.3 USER STORIES



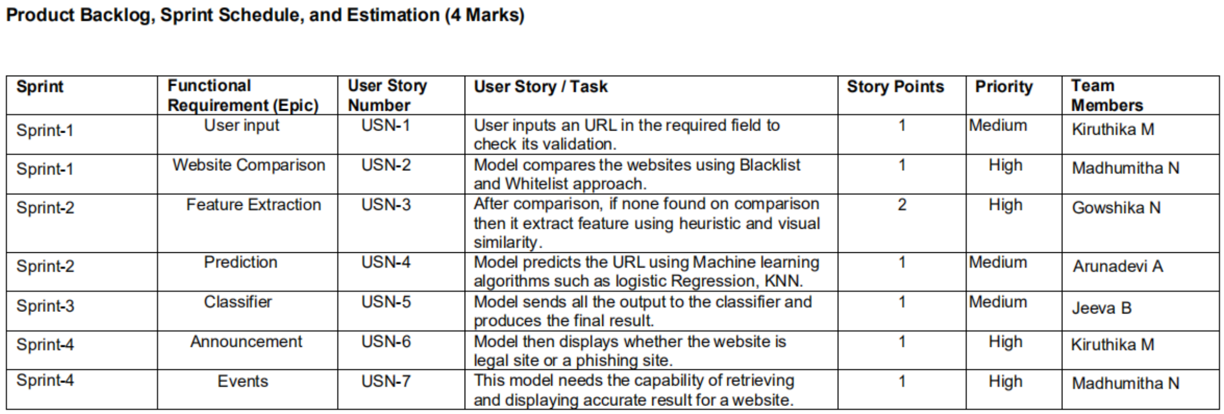
User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Gmail	I can register & access the dashboard with Gmail Login	Medium	Sprint-2
	Login	USN-4	As a user, I can log into the application by entering email & password	I can access the website features	High	Sprint-2
	User Input	USN-5	As a user, I can input the URL in the required field and wait for validation	I can access the detailed result of the URL	High	Sprint-3
Administrator	Data Collection	USN-6	The data to identify the phishing link is to be collected	The model is ready to train	High	Sprint-3
	Data Pre-Processing	USN-7	The data is to be cleaned to provide better accuracy	The model is ready with high accuracy	High	Sprint-4
	Model Deployment	USN-8	The trained and tested model is deployed using the Machine learning algorithm	I have the model which is successfully deloyed	High	Sprint-5
	Application Building	USN-9	As a admin, The user page must be designed to access the feature in more ease manner	I have the live website	High	Sprint-5

6. PROJECT PLANNING & SCHEDULING

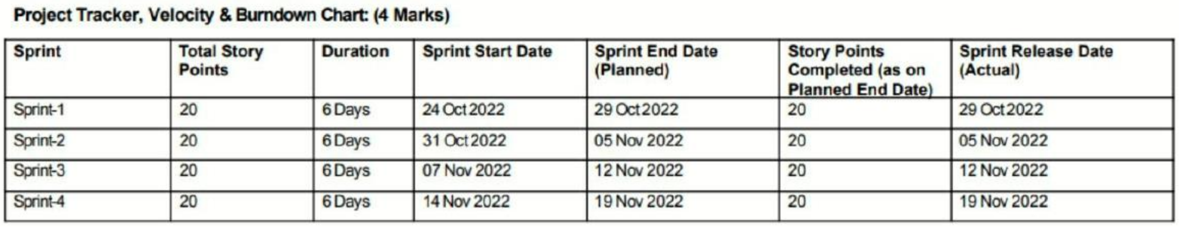
6.1 SPRINT PLANNING AND ESTIMATION:



Product Backlog, Sprint Schedule, and Estimation (4 Marks)

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	Kiruthika M
Sprint-1	Website Comparison	USN-2	Model compares the websites using Blacklist and Whitelist approach.	1	High	Madhumitha N
Sprint-2	Feature Extraction	USN-3	After comparison, if none found on comparison then it extract feature using heuristic and visual similarity.	2	High	Gowshika N
Sprint-2	Prediction	USN-4	Model predicts the URL using Machine learning algorithms such as logistic Regression, KNN.	1	Medium	Arunadevi A
Sprint-3	Classifier	USN-5	Model sends all the output to the classifier and produces the final result.	1	Medium	Jeeva B
Sprint-4	Announcement	USN-6	Model then displays whether the website is legal site or a phishing site.	1	High	Kiruthika M
Sprint-4	Events	USN-7	This model needs the capability of retrieving and displaying accurate result for a website.	1	High	Madhumitha N

6.2 SPRINT DELIVERY SCHEDULE:



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	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 AND SOLUTIONING:

7.1 FEATURE 1

Long URL: Long URL's can be used to hide the suspicious part of in the address bar. Although scientifically there is reliable method of predicting the range of length that justify a website as phishing or non-phishing but then it is criteria used with other features in detecting suspicious sites. In the study of Basnet et al. (2011), a proposed length of ≤75 but there was no justification for behind their value. In this project a URL length of >127 character is used for non-phishing and ≤127 character for phishing website. This value is chosen based on the dataset collected by manually comparing the length of the most lengthy non-phishing website and phishing website in the dataset.

7.2 FEATURE 2

At "@" symbol: The phishing URL may include the "@" symbol somewhere within the address because the web browser, when reading an internet address; ignore everything to the left of the @ symbol, therefore, the address ebay.com@fake-auction.com would actually be "fake-auction.com."

Hexadecimal: Particular to phishing are hex-encoded URLs. In the interest of compatibility, most mail user agents, web browsers, and HTTP servers all understand basic hex-encoded character equivalents, so that: http://210.219.241.125/images/paypal/cgi-bin/webscrcmd_login.php and http://%32%31%30.%32%31%39%2e%32%34%31%2e%31%32%35/%69%6d%61%67%65%73/paypal/cgi-bin/webscrcmd_login.php are functionally equivalent. The main illicit purpose of this encoding is to evade blacklist-based anti-spam filters which do not process hex character encoding (effectively, another insertion attack). It also evades protection mechanisms that prohibit IP addresses as URL destinations, on the assumption that "normal" http links will use more familiar DNS names.

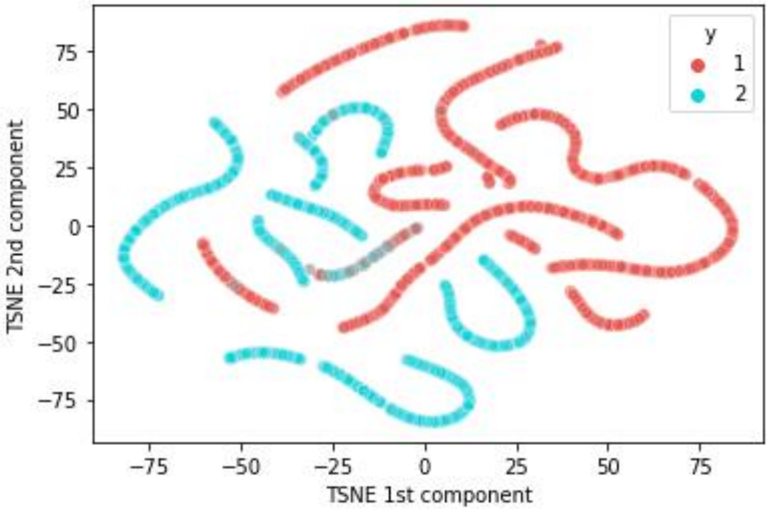
7.3 DATABASE SCHEME:

Page’s souce code based features: Includes URLs embedded in the webpage and HTML and Javascript based features.

Domain based features

URL and derived features

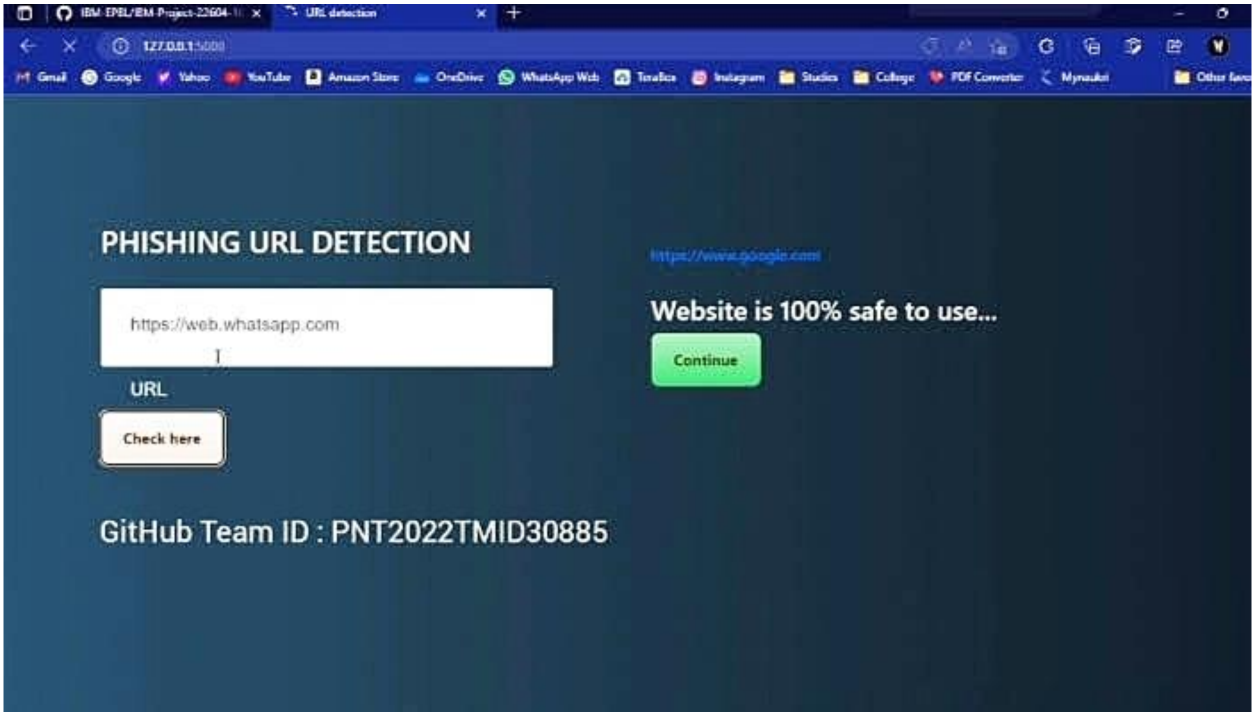
Studying the way of extraction and relevance of features, we dropped 5 features out of 30, namely: Port Number, Abnormal URL, Pop-up Window, Google Index and Number of Links Pointing to a Page. Port Number was dropped due to feature drift. Rest were dropped due to unavailability of methods to extract them programmatically or absence of public APIs.



TSNE plot (1 represents Legitimate URLs and 2 represents Phishing URLs)

8. TESTING:

8.1 TEST CASE:



PHISHING URL DETECTION

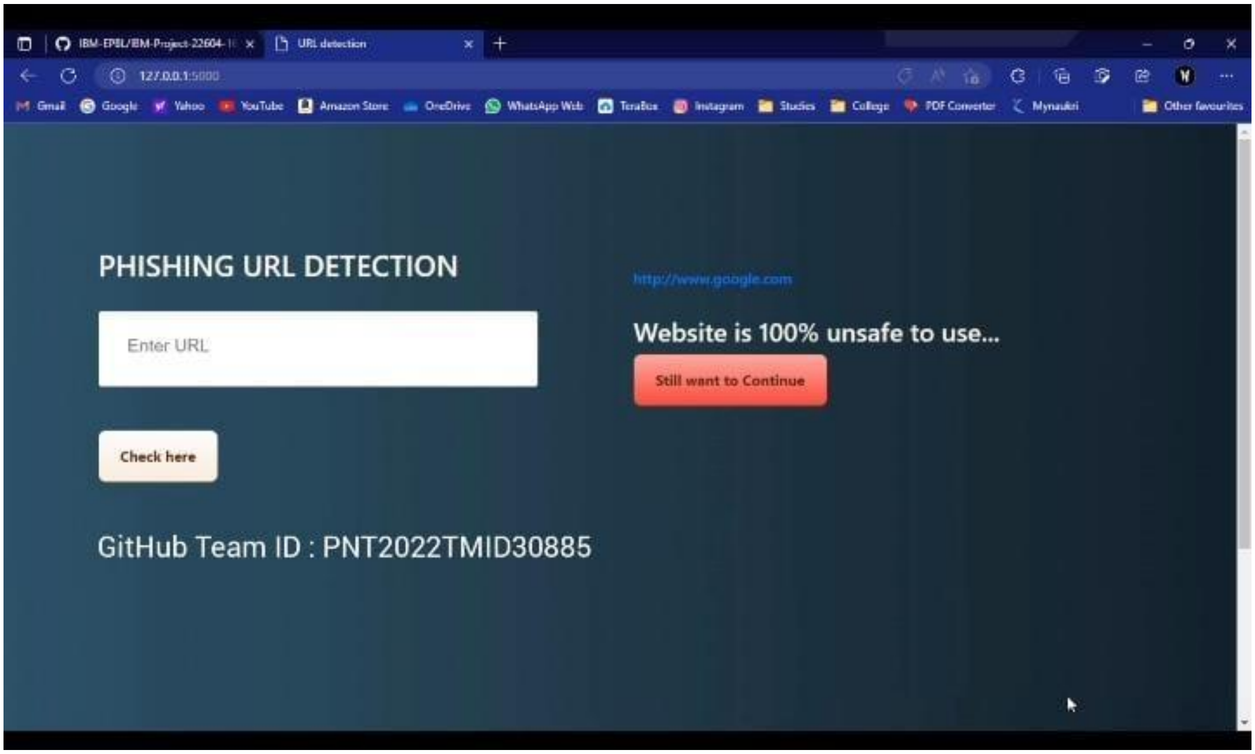
https://web.whatsapp.com

URL

https://www.google.com

Website is 100% safe to use...

GitHub Team ID : PNT2022TMID30885



PHISHING URL DETECTION

Enter URL

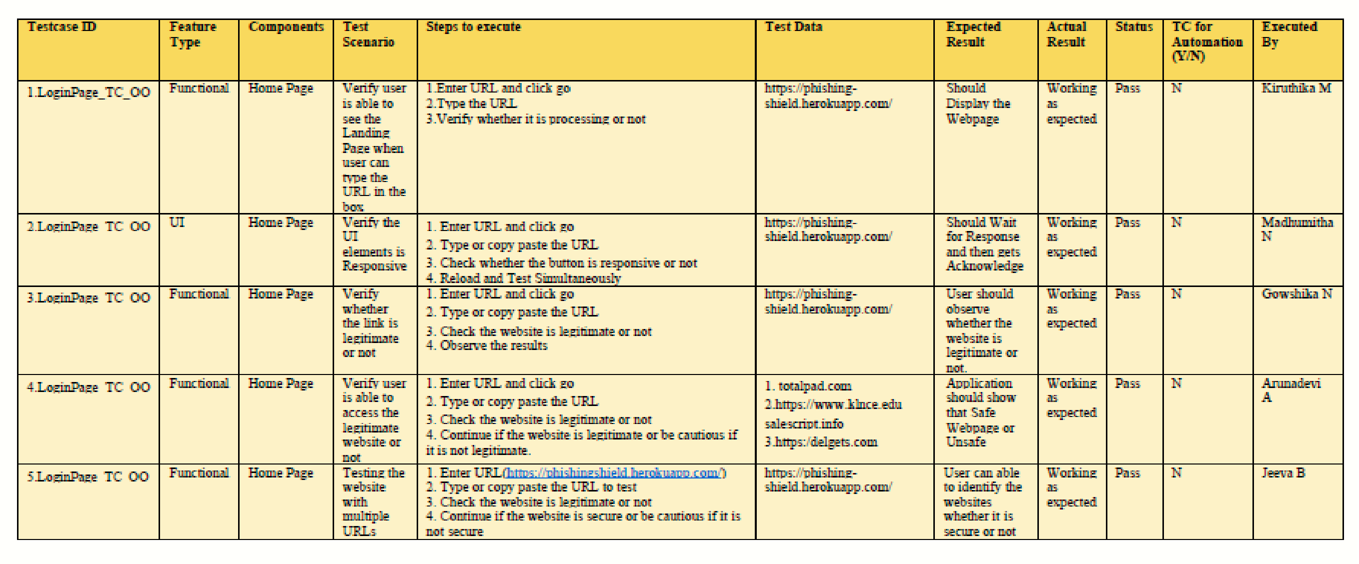
http://www.google.com

Website is 100% unsafe to use...

GitHub Team ID : PNT2022TMID30885

8.2 USER ACCEPTANCE TESTING:

Performance Testing



Testcase ID	Feature Type	Components	Test Scenario	Steps to execute	Test Data	Expected Result	Actual Result	Status	TC for Automation (Y/N)	Executed By
1.LoginPage_TC_OO	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	Kiruthika M
2.LoginPage_TC_OO	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	Madhumitha N
3.LoginPage_TC_OO	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	Gowshika N
4.LoginPage_TC_OO	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.	1. totalpad.com 2.https://www.klnce.edu salescript.info 3.https:/delgets.com	Application should show that Safe Webpage or Unsafe	Working as expected	Pass	N	Arunadevi A
5.LoginPage_TC_OO	Functional	Home Page	Testing the website with multiple URLs	1. Enter URL(https://phishingshield.herokuapp.com/) 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	https://phishing-shield.herokuapp.com/	User can able to identify the websites whether it is secure or not	Working as expected	Pass	N	Jeeva B

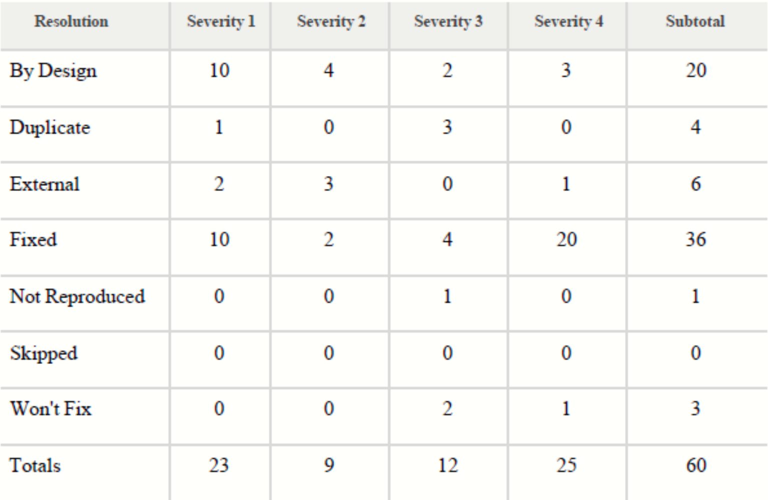
Acceptance Testing

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

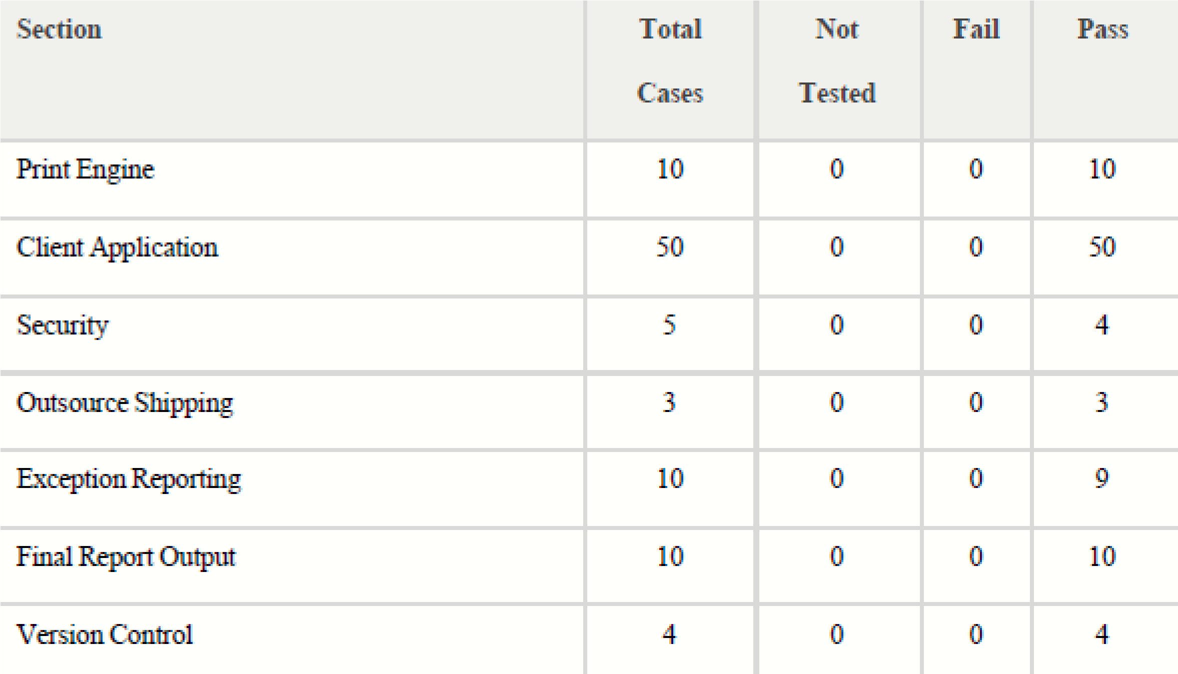
This report shows the number of resolved or closed bugs at each severity level, and how they were resolved.



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

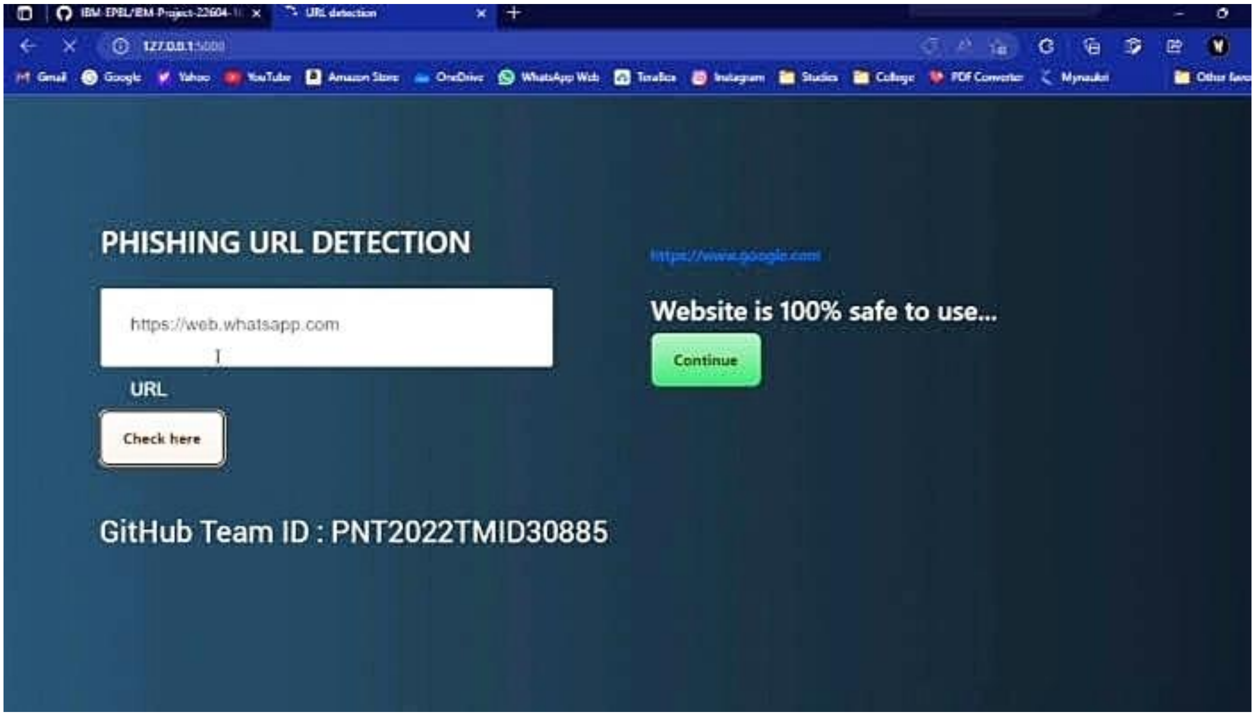
3. Test Case Analysis

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. RESULT:



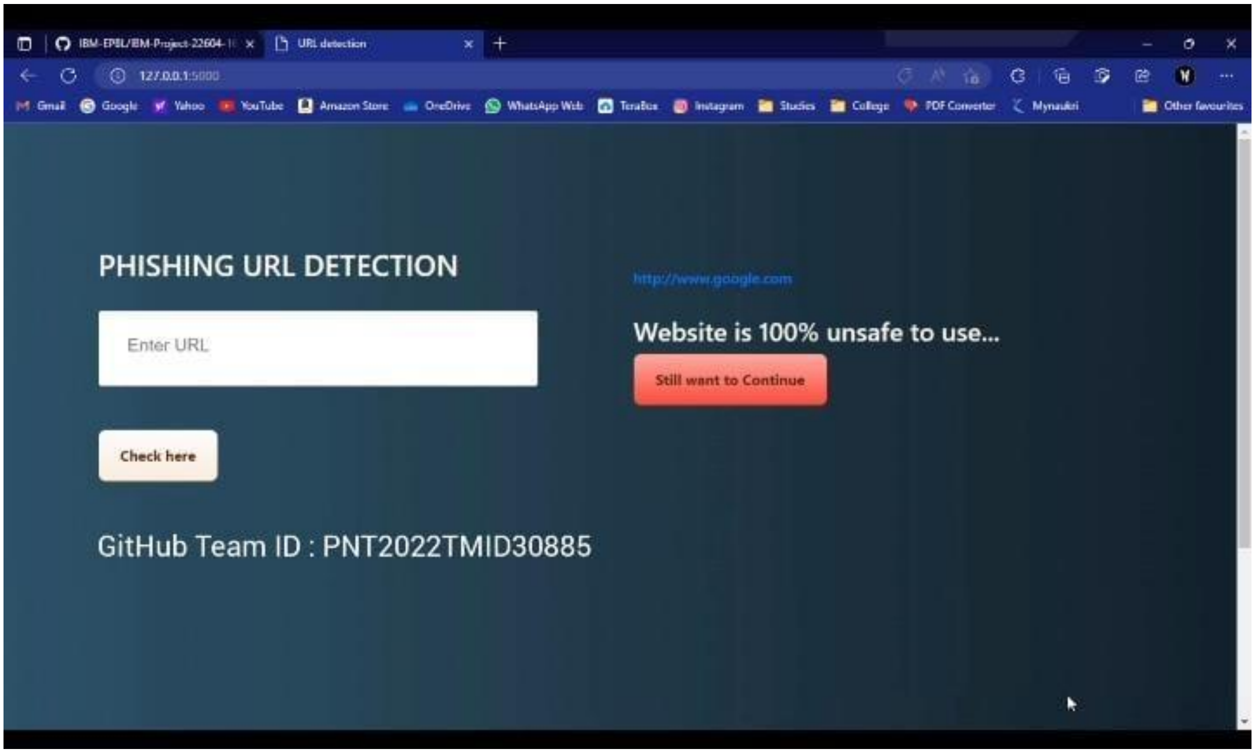
PHISHING URL DETECTION

https://web.whatsapp.com

https://www.google.com

Website is 100% safe to use...

GitHub Team ID : PNT2022TMID30885



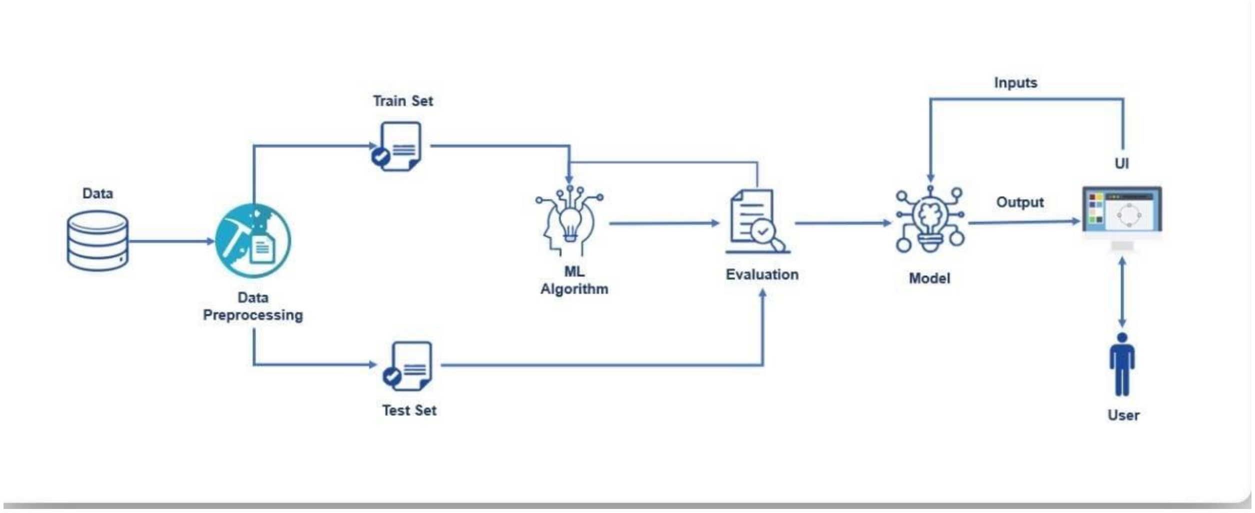
PHISHING URL DETECTION

http://www.google.com

Website is 100% unsafe to use...

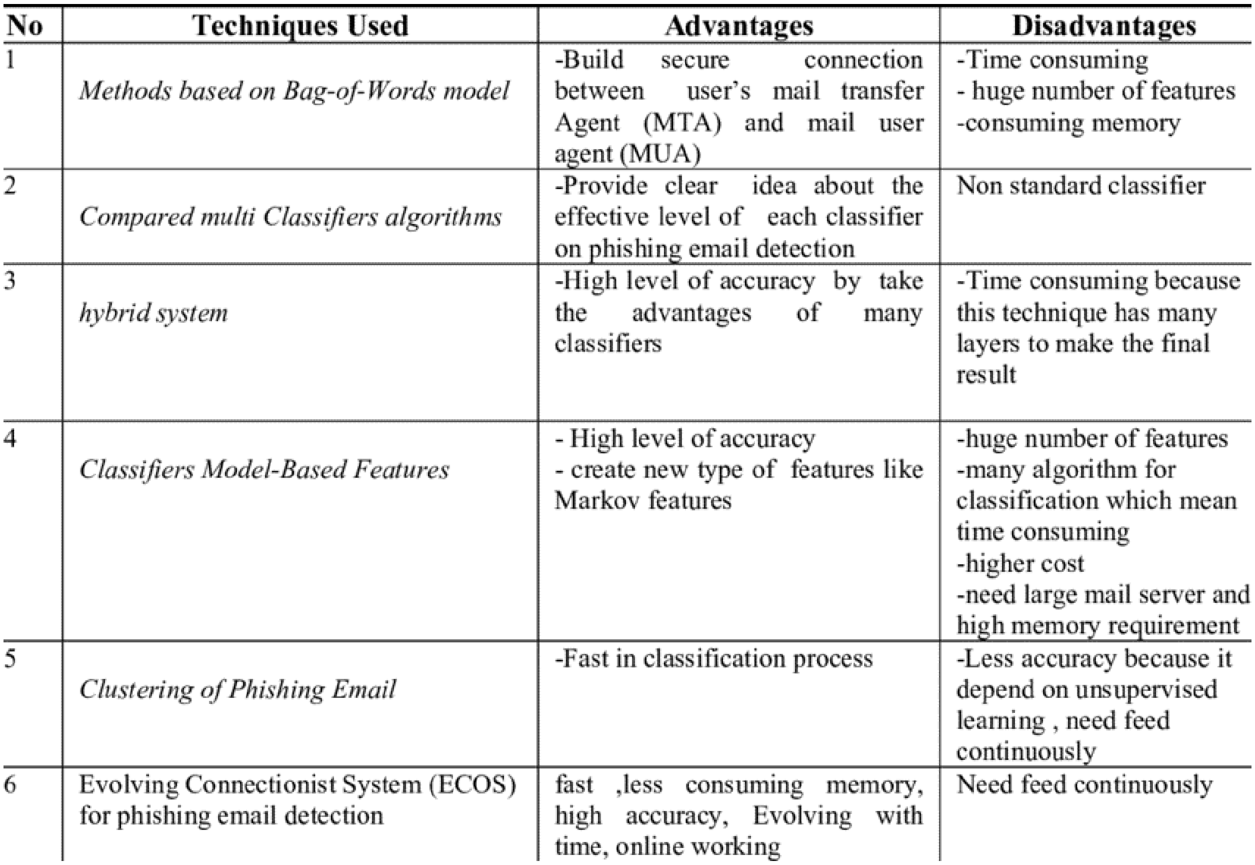
GitHub Team ID : PNT2022TMID30885

9.1 PERFORMANCE METRICS:



Phishing emails can reach millions of users directly, and hide amongst the huge number of benign emails that busy users receive. Attacks can install malware (such as ransomware), sabotage systems, or steal intellectual property and money.

10.ADVANTAGES AND DISADVANTAGES:



No	Techniques Used	Advantages	Disadvantages
1	*Methods based on Bag-of-Words model*	-Build secure connection between user's mail transfer Agent (MTA) and mail user agent (MUA)	-Time consuming - huge number of features -consuming memory
2	*Compared multi Classifiers algorithms*	-Provide clear idea about the effective level of each classifier on phishing email detection	Non standard classifier
3	*hybrid system*	-High level of accuracy by take the advantages of many classifiers	-Time consuming because this technique has many layers to make the final result
4	*Classifiers Model-Based Features*	- High level of accuracy - create new type of features like Markov features	-huge number of features -many algorithm for classification which mean time consuming -higher cost -need large mail server and high memory requirement
5	*Clustering of Phishing Email*	-Fast in classification process	-Less accuracy because it depend on unsupervised learning , need feed continuously
6	Evolving Connectionist System (ECOS) for phishing email detection	fast ,less consuming memory, high accuracy, Evolving with time, online working	Need feed continuously

11. CONCLUSION :

This paper aims to enhance detection method to detect phishing websites using machine learning technology. We achieved 97.14% detection accuracy using random forest algorithm with lowest false positive rate. Also result shows that classifiers give better performance when we used more data as training data. In future hybrid technology will be implemented to detect phishing websites more accurately, for which random forest algorithm of machine learning technology and blacklist method will be used.

12. FUTURE SCOPE:

This paper presented an intelligent phishing detection and protection scheme by employing a new approach using the integrated features of images, 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.

13. APPENDIX:

SOURCE CODE:

App Python 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)
```

IBM App – Python 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
import requests
# NOTE: you must manually set API_KEY below using information retrieved
fromyour IBM Cloud account.
API_KEY = "l4YwAYfz6xcdukNsk8cMF3WDQrAlcdT9xVUs6QxsG87-"
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}

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)
# NOTE: manually define and pass the array(s) of values to be scored in the nextline
payload_scoring = {"input_data": [{"fields":
[['f0','f1','f2','f3','f4','f5','f6','f7','f8','f9','f10','f11','f12','f13','f14','f15','f16','f17','f18','f
19','f20','f21','f22','f23','f24','f25','f26','f27','f28','f29']], "values": [[-1,1,1,1,-1,-1,-1,-
1,-1,1,1,-1,1,-1,1,-1,-1,-1,0,1,1,1,1,-1,-1,-1,-1,1,1,-1]]}]}
response_scoring = requests.post('https://us-
south.ml.cloud.ibm.com/ml/v4/deployments/7bc163a4-37f7-4ec6-b0bf-
63d1e40a85e4/predictions?version=2022-11-16', json=payload_scoring,
headers={'Authorization': 'Bearer ' + mltoken})
#print("Scoring response")
#print(response_scoring.json())
pred=response_scoring.json()
```

```
output=pred['predictions'][0]['values'][0][0]
y_pred =output
#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 gg{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)
# NOTE: manually define and pass the array(s) of values to be scored in the nextline
payload_scoring = {"input_data": [{"fields":
[['f0','f1','f2','f3','f4','f5','f6','f7','f8','f9','f10','f11','f12','f13','f14','f15','f16','f17','f18','f
19','f20','f21','f22','f23','f24','f25','f26','f27','f28','f29']], "values": [[-1,1,1,1,-1,-1,-1,-
1,-1,1,1,-1,1,-1,1,-1,-1,-1,0,1,1,1,1,-1,-1,-1,-1,1,1,-1]]}]}
response_scoring = requests.post('https://us-
south.ml.cloud.ibm.com/ml/v4/deployments/7bc163a4-37f7-4ec6-b0bf-
63d1e40a85e4/predictions?version=2022-11-16', json=payload_scoring,
headers={'Authorization': 'Bearer ' + mltoken})
#print("Scoring response")
#print(response_scoring.json())
pred=response_scoring.json()

output=pred['predictions'][0]['values'][0][0]
```

GITHUB & PROJECT DEMO LINK:

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-38431-1660380573

PROJECT DEMO LINK:

https://drive.google.com/file/d/1UMr6uCiEXmrJxWuV8zrUhTg-fQtU_7tH/view?usp=share_link