Project report on

Web Phising Detection

TEAM ID: PNT2022TMID31980

M.KEERTHIRAJAN	B.E(CSE) - [731619104024]
P.BARATH	B.E(CSE) - [731619104008]
M.MADHUMITHA	B.E(CSE) - [731619104027]
M.JEEVA	B.E(CSE) - [731619104020]
K.PAVITHRA	B.E(CSE) - [731619104040]

CONTENT

1. INTRODUCTION

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.

1.2 PURPOSE

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.

2.LITERATURE SURVEY

2.1 EXISTING PROBLEM

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. 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.

2.2 REFERENCES

- LongfeiWu etal..., "Effective Defense Schemes for Phishing Attacks on Mobile

Computing Platforms, " IEEE_2016, pp.6678-6691.

- Surbhi Gupta etal., "ALiterature Survey on Social Engineering Attacks: PhishingAttacks," in International Conference on Computing,Communication and Automation(ICCCA2016),2016, pp. 537-540.
- Guardian Analytics, "APractical Guide to AnomalyDetection Implications of meeting new FFIEC minimum expectations for layered security". [Accessed : 08 Jan2015]
- SANS Institute, "Phishing : AnAnalysis of a GrowingProblem",2007.1417[Accessed : 23 May 2017]
- Phys.: Conf. Ser. "A literature survey on Retraction: Phishing website detection using machine Learning and deep learning techniques" 1916 (2021)012407.

2.3 PROBLEM STATEMENT DEFINITION

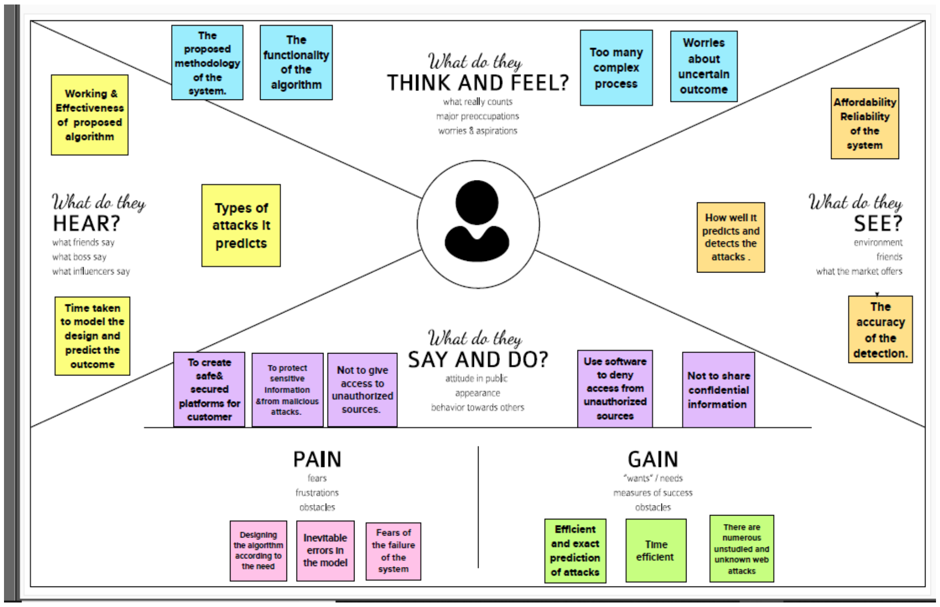
Phishing detection techniques do suffer low detection accuracy and high false alarm especially when novel phishing approaches are introduced. Besides, the most common technique used, blacklist-based method is inefficient in responding to emanating phishing attacks since registering new domain has become easier, no comprehensive blacklist can ensure a perfect up-to-date database. Furthermore, page content inspection has been used by some strategies to overcome the false negative problems and complement the vulnerabilities of the stale lists. Moreover, page content inspection algorithms each have different approach to phishing website detection with varying degrees of accuracy.

3.IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

An empathy map is a collaborative visualization used to articulate what we know about a particular type of user. It externalizes knowledge about users in order to 1) create a shared understanding of user needs, and 2) aid in decision making.

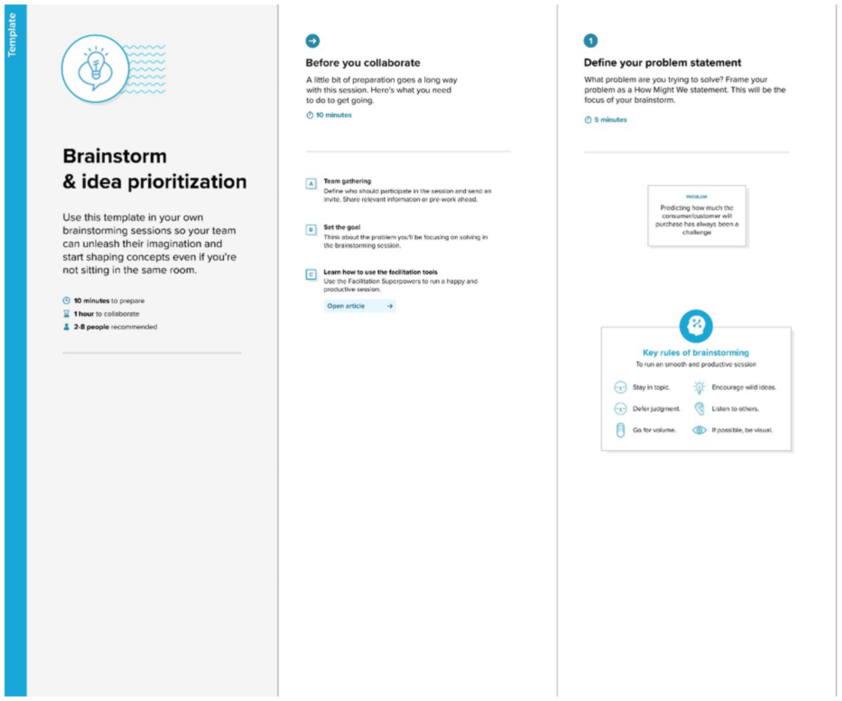
Traditional empathy maps are split into 4 quadrants (Says, Thinks, Does, and Feels), with the user or persona in the middle. Empathy maps provide a glance into who a user is as a whole and are not chronological or sequential.


What do they
THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
The proposed methodology of the system.
The functionality of the algorithm
Too many complex process
Worries about uncertain outcome
What do they
HEAR?
what friends say
what boss say
what influencers say
Working & Effectiveness of proposed algorithm
Types of attacks it predicts
Time taken to model the design and predict the outcome
What do they
SEE?
environment
friends
what the market offers
Affordability Reliability of the system
How well it predicts and detects the attacks .
The accuracy of the detection.
What do they
SAY AND DO?
attitude in public
appearance
behavior towards others
To create safe& secured platforms for customer
To protect sensitive information &from malicious attacks.
Not to give access to unauthorized sources.
Use software to deny access from unauthorized sources
Not to share confidential information
PAIN
fears
frustrations
obstacles
Designing the algorithm according to the need
Inevitable errors in the model
Fears of the failure of the system
GAIN
"wants" / needs
measures of success
obstacles
Efficient and exact prediction of attacks
Time efficient
There are numerous unstudied and unknown web attacks

3.2 Ideation & Brainstorming

Brainstorming is a method design teams use to generate ideas to solve clearly defined design problems. In controlled conditions and a free-thinking environment, teams approach a problem by such means as "How Might We" questions.

Step-1: Team Gathering, Collaboration and Select the ProblemStatement



Template

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.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

1 **Define your problem statement**

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

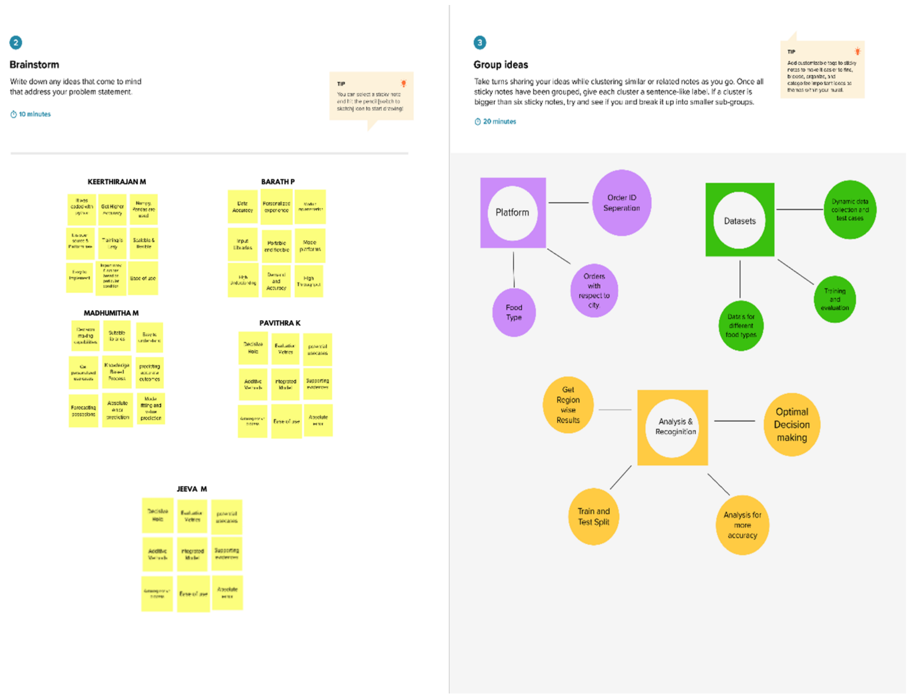
5 minutes

PROBLEM
Predicting how much the consumer/customer will purchase has always been a challenge

Key rules of brainstorming
To run an smooth and productive session

Stay in topic.
Encourage wild ideas.
Defer judgment.
Listen to others.
Go for volume.
If possible, be visual.

Step-2: Brainstorm, Idea Listing and Grouping



2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

3

Group ideas

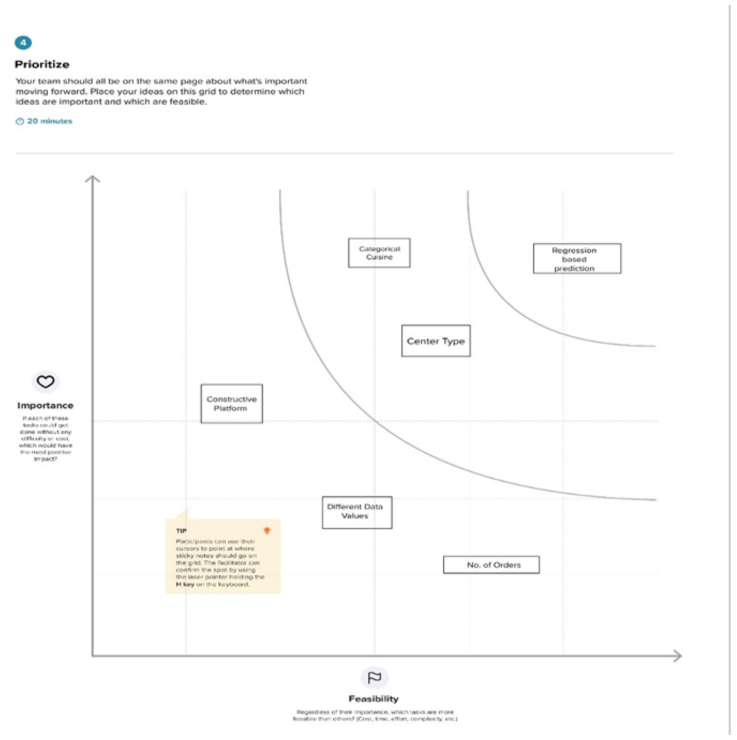
Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

Step-3: Idea Prioritization



4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

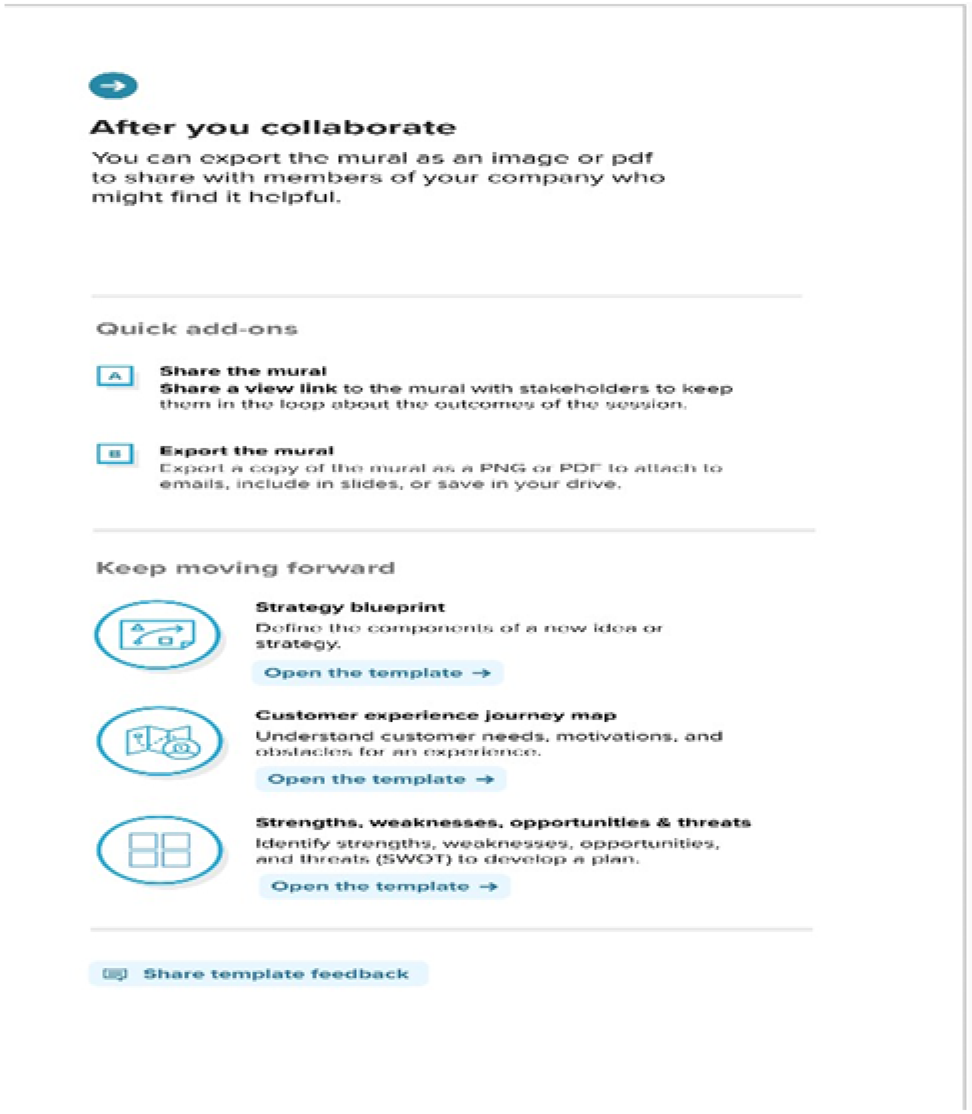
20 minutes

TIP

Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the H key on the keyboard.

Categorical Cuisine
Regression based prediction
Center Type
Constructive Platform
Different Data Values
No. of Orders
Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?
Feasibility
Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

Step-4 :



After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

A **Share the mural**
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.

B **Export the mural**
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

Strategy blueprint
Define the components of a new idea or strategy.
Open the template →

Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.
Open the template →

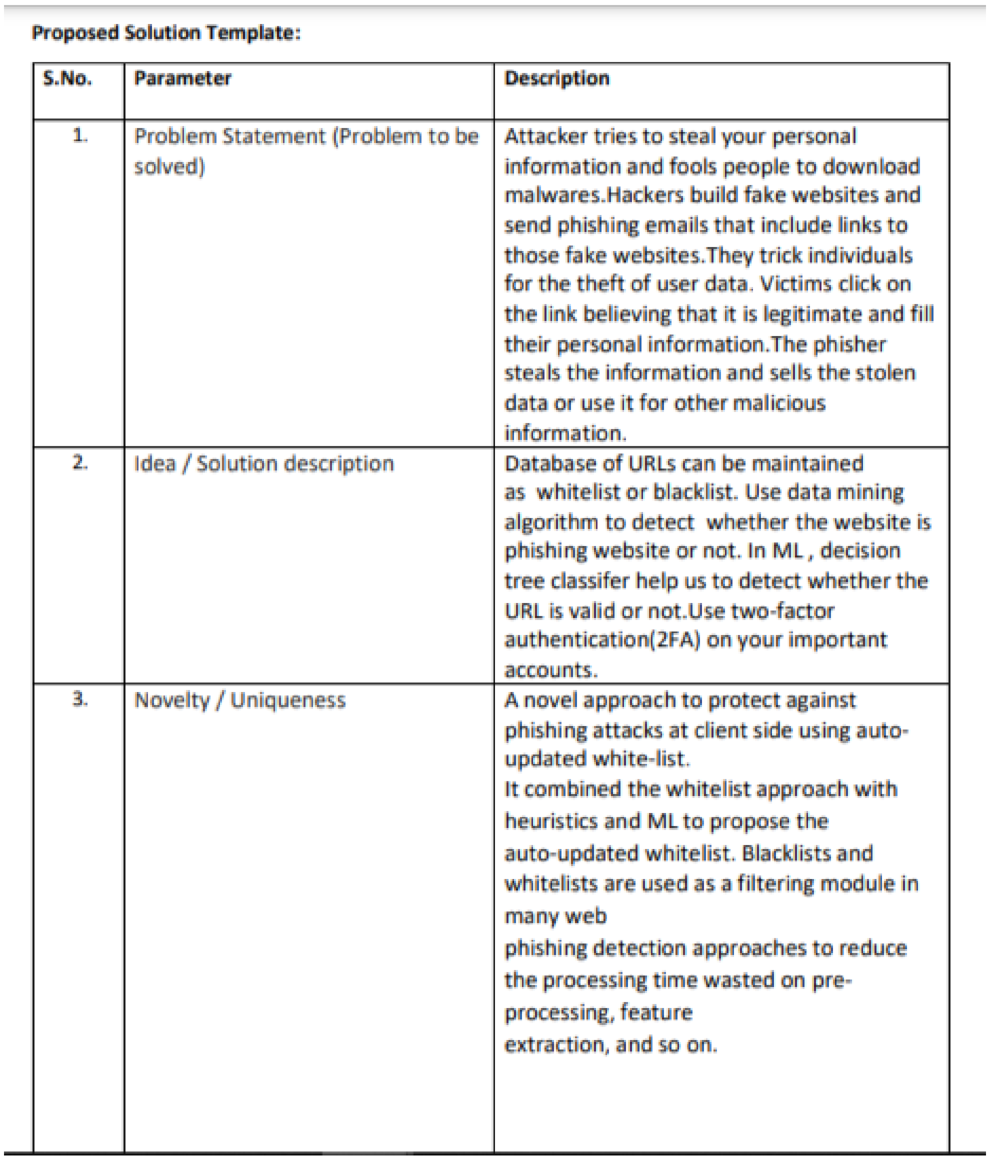
Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
Open the template →

Share template feedback

3.3 Proposed Solution

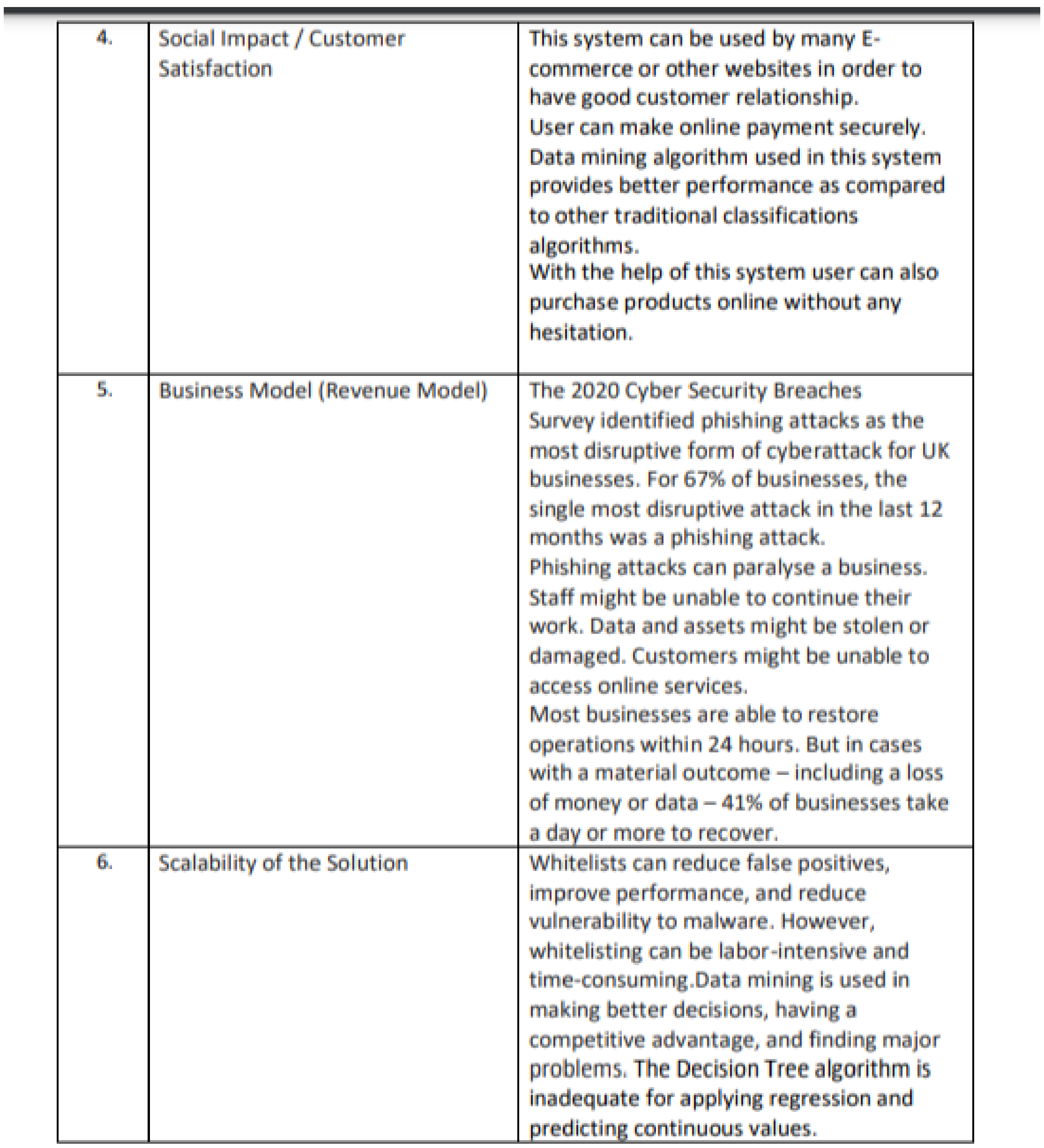
Proposed Solution means the technical solution to be provided by the Implementation agency in response to the requirements and the objectives of the Project.Proposed Solution means the Proposed System with modifications that meet the Agency’s requirements as set forth in this RFP.Proposed Solution means the combination of software, hardware, other products or equipment, and any and all services (including any installation, implementation, training, maintenance and support services) necessary to implement the solution described by Vendor in its Proposal.Create a problem statement to understand your customer's point of view.

The Customer Problem Statement template helps you focus on what matters to create experiences people will love. A well-articulated customer problem statement allows you and your team to find the ideal solution for the challenges your customers face. Throughout the process, you'll also be able to empathize with your customers, which helps you better understand how they perceive your product or service.



Proposed Solution Template:

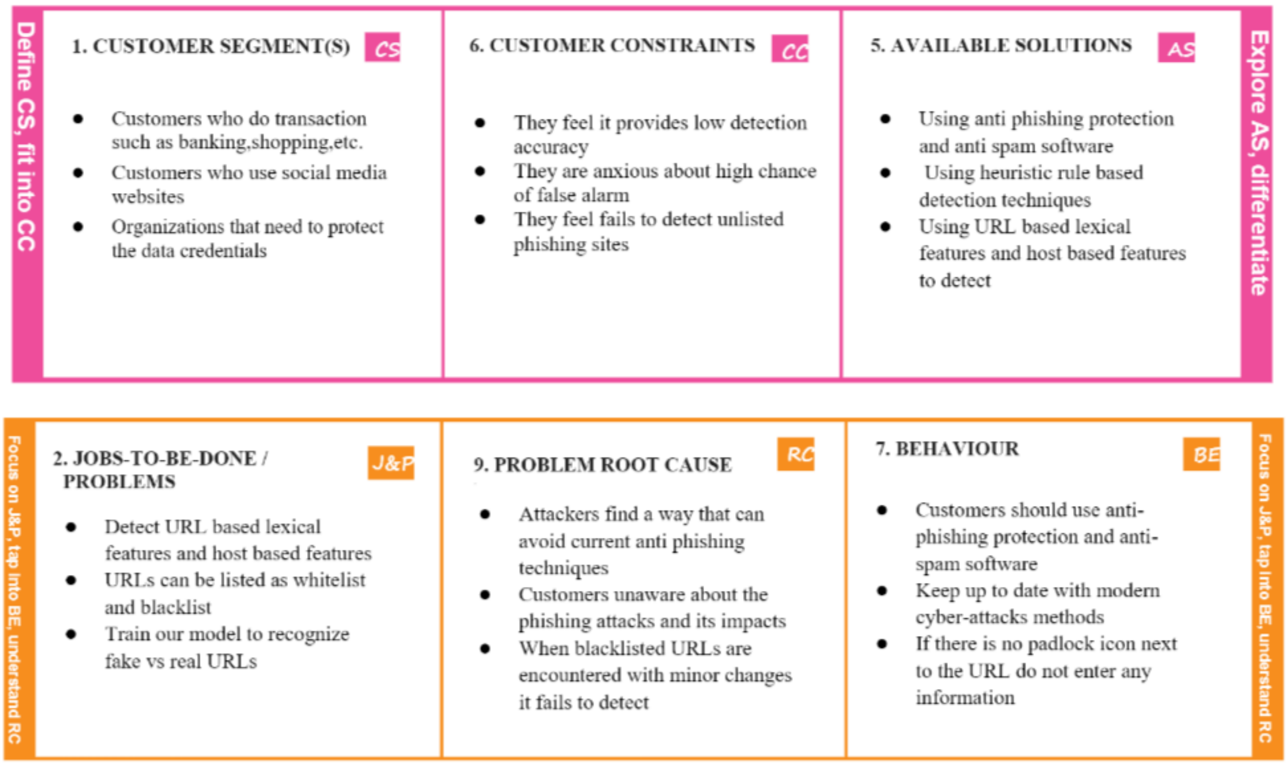
S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Attacker tries to steal your personal information and fools people to download malwares.Hackers build fake websites and send phishing emails that include links to those fake websites.They trick individuals for the theft of user data. Victims click on the link believing that it is legitimate and fill their personal information.The phisher steals the information and sells the stolen data or use it for other malicious information.
2.	Idea / Solution description	Database of URLs can be maintained as whitelist or blacklist. Use data mining algorithm to detect whether the website is phishing website or not. In ML , decision tree classifer help us to detect whether the URL is valid or not.Use two-factor authentication(2FA) on your important accounts.
3.	Novelty / Uniqueness	A novel approach to protect against phishing attacks at client side using auto-updated white-list. It combined the whitelist approach with heuristics and ML to propose the auto-updated whitelist. Blacklists and whitelists are used as a filtering module in many web phishing detection approaches to reduce the processing time wasted on pre-processing, feature extraction, and so on.



4.	Social Impact / Customer Satisfaction	This system can be used by many E-commerce or other websites in order to have good customer relationship. User can make online payment securely. Data mining algorithm used in this system 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.
5.	Business Model (Revenue Model)	The 2020 Cyber Security Breaches Survey identified phishing attacks as the most disruptive form of cyberattack for UK businesses. For 67% of businesses, the single most disruptive attack in the last 12 months was a phishing attack. Phishing attacks can paralyse a business. Staff might be unable to continue their work. Data and assets might be stolen or damaged. Customers might be unable to access online services. Most businesses are able to restore operations within 24 hours. But in cases with a material outcome – including a loss of money or data – 41% of businesses take a day or more to recover.
6.	Scalability of the Solution	Whitelists can reduce false positives, improve performance, and reduce vulnerability to malware. However, whitelisting can be labor-intensive and time-consuming.Data mining is used in making better decisions, having a competitive advantage, and finding major problems. The Decision Tree algorithm is inadequate for applying regression and predicting continuous values.

3.4 Problem Solution fit

Proposed Solution means the technical solution to be provided by the Implementation agency in response to the requirements and the objectives of the Project.Proposed Solution means the Proposed System with modifications that meet the Agency's requirements as set forth in this RFP.Proposed Solution means the combination of software, hardware, other products or equipment, and any and all services (including any installation, implementation, training, maintenance and support services) necessary to implement the solution described byVendorinits Proposal.



Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

- Customers who do transaction such as banking,shopping,etc.
- Customers who use social media websites
- Organizations that need to protect the data credentials

6. CUSTOMER CONSTRAINTS CC

- They feel it provides low detection accuracy
- They are anxious about high chance of false alarm
- They feel fails to detect unlisted phishing sites

5. AVAILABLE SOLUTIONS AS

- Using anti phishing protection and anti spam software
- Using heuristic rule based detection techniques
- Using URL based lexical features and host based features to detect

Explore AS, differentiate

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

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

- Detect URL based lexical features and host based features
- URLs can be listed as whitelist and blacklist
- Train our model to recognize fake vs real URLs

9. PROBLEM ROOT CAUSE RC

- Attackers find a way that can avoid current anti phishing techniques
- Customers unaware about the phishing attacks and its impacts
- When blacklisted URLs are encountered with minor changes it fails to detect

7. BEHAVIOUR BE

- Customers should use anti-phishing protection and anti-spam software
- Keep up to date with modern cyber-attacks methods
- If there is no padlock icon next to the URL do not enter any information

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

4. REQUIREMENT ANALYSIS

4.1 Functional system requirement

Extension plugin should provide a warning pop-up when they visit a website that is phished; therefore it should strictly follow the following:

a. Extension plugin ability to present the pop-up to the users screen should be quick enough to the point, users will be aware before entering any confidential or sensitive details into a phishing website.

b. Extension plugin should not need the facilities and services from an 3rd party service or APIs, due the reason that those services will always the potential to leak users browsing data and pattern when it gets compromised by hackers

c. Extension plugin will have the capability to also detect latest and new phishing websites

4.2 Non-Functional requirements

Graphical User Interface design Interface developed should be done with the understanding that it must meet the simplicity of what users would like to see when they need an extension for detecting things, and also it needs to adhere to non IT literate users as well. It must also provide the exact information on what the user wants like identifying a phishing website quickly without needing to click on many options. The process of identifying phishing website should be taken directly from the web-page user wants to view through their URL and the result from it should be easily understood by the users. Most importantly, the extension plugin should have a popup that will notify the user regarding the website status of being phished.

Software requirements:

a. PyCharm software

b. Python language

c. Google chrome browser

d. Scikit-learn

e. NumPy

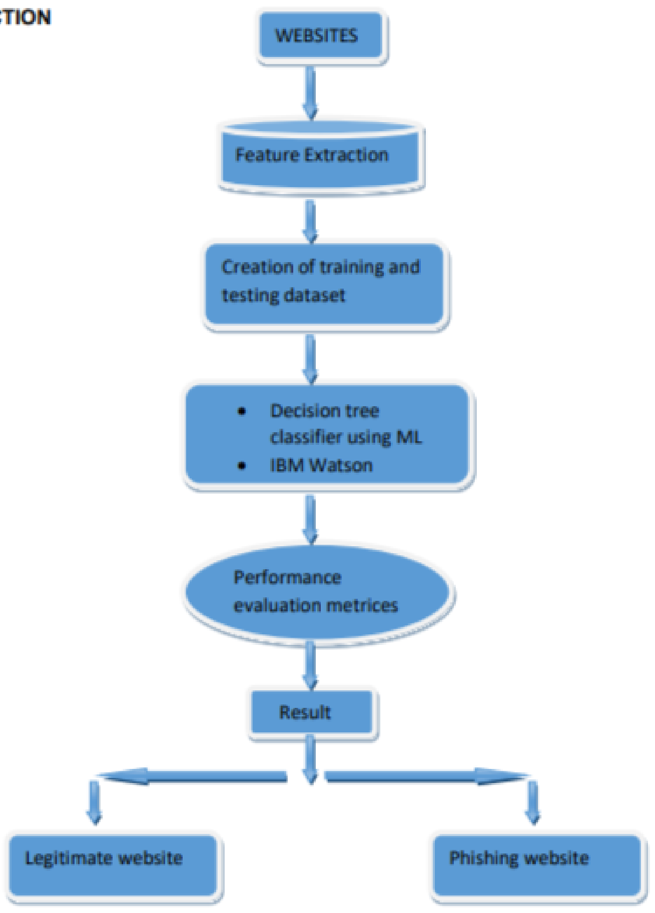
f. Liac-arff for dataset

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.

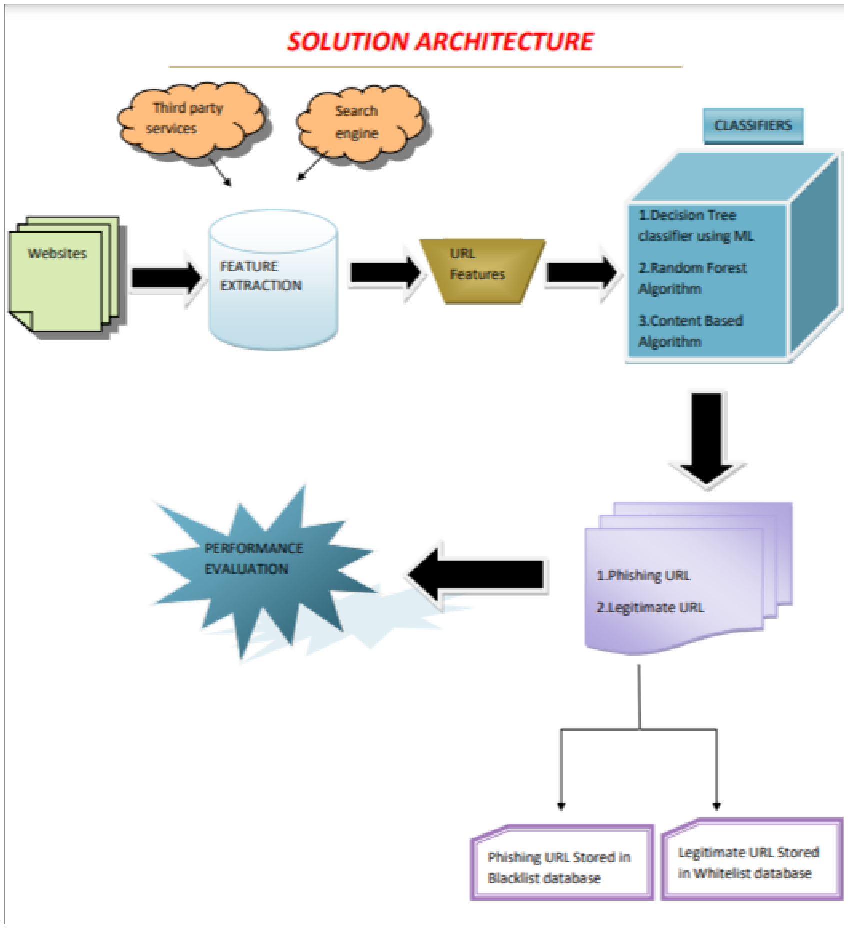
WEB PHISHING DETECTION


WEBSITES
Feature Extraction
Creation of training and testing dataset
• Decision tree classifier using ML
• IBM Watson
Performance evaluation metrices
Result
Legitimate website
Phishing website

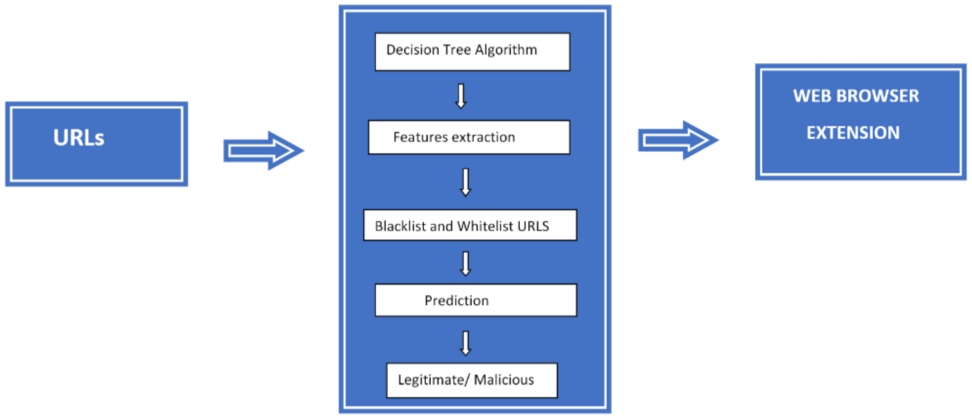
5.2 Solution & Technical Architecture

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 decison.


SOLUTION ARCHITECTURE
Third party services
Search engine
CLASSIFIERS
Websites
FEATURE EXTRACTION
URL Features
1.Decision Tree classifier using ML
2.Random Forest Algorithm
3.Content Based Algorithm
PERFORMANCE EVALUATION
1.Phishing URL
2.Legitimate URL
Phishing URL Stored in Blacklist database
Legitimate URL Stored in Whitelist database

TECHNICAL ARCHITECHTURE



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 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 Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can register & access the dashboard with Gmail Login	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can access my dashboard	High	Sprint-1
	Dashboard	USN-6	As a user, I can access the dashboard to get information	I can access my application	High	Sprint-1
Customer (Web user)	Registration	USN-7	As a web user, I can register my details in official websites and I will create strong passwords	I can access my dashboard/account safely	High	Sprint-1
	Login & Dashboard		As a web user, I can login into application by using my user id and password	I can access the resources	High	Sprint-1
Customer Care Executive	Login	CCE-1	As a CCE I can login to website using user id and password and I can interact with the user	I can access the website	High	Sprint-1
	Dashboard	CCE-2	As a CCE I can login to dashboard using user id and password and I can interact with the user and I can explain the app usage and rectify their issues.	I can access the resources	High	Sprint-1
Administrator	Login & Dashboard	A-1	As an administrator, I can access the dashboard and direct activities.	I will maintain the database safely	High	Sprint-1

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Homepage	USN-1	As a user, I can explore the resources of the homepage for the functioning	10	Low	Keerthirajan.M,barath.p
Sprint-1		USN-2	As a user, I can learn about the various sides of the web phishing and be aware of the scams	5	High	Madhumitha.M, Jeeva.M
Sprint-2	Final page	USN-3	As a user, I can explore the resources of the final page for the functioning	15	Low	Keerthirajan.M, Barath.P
Sprint-3	Prediction	USN-4	As a user, I can predict the URL easily for detecting whether the website is legitimate or not	10	High	Madhumitha.M, Jeeva.M, Pavithra.k
Sprint-4	Chat	USN-5	As a user, I can share the experience or contact the admin for the support	10	High	Madhumitha.M, Jeeva.M, Pavithra.k,Barath.P
Sprint-1	Homepage	USN-6	As a admin, we can design interface and maintain the functioning of the website	5	High	Keerthirajan.M, Barath.P
Sprint-2	Final page	USN-7	As a admin, we can design the complexity of the website for making it user-friendly	5	Medium	Pavithra.K, Jeeva.M
Sprint-3	Prediction	USN-8	As a admin, we can use various ML classifier model for the accurate result for the detection of URL	10	High	Keerthirajan.M, Barath.P,Jeeva.M, Madhumitha.M
Sprint-4	Chat	USN-9	As a admin, we can response to the user message for improvement of the website	10	Medium	Keerthirajan.M, Madhumitha.M

6.2 Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	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	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)

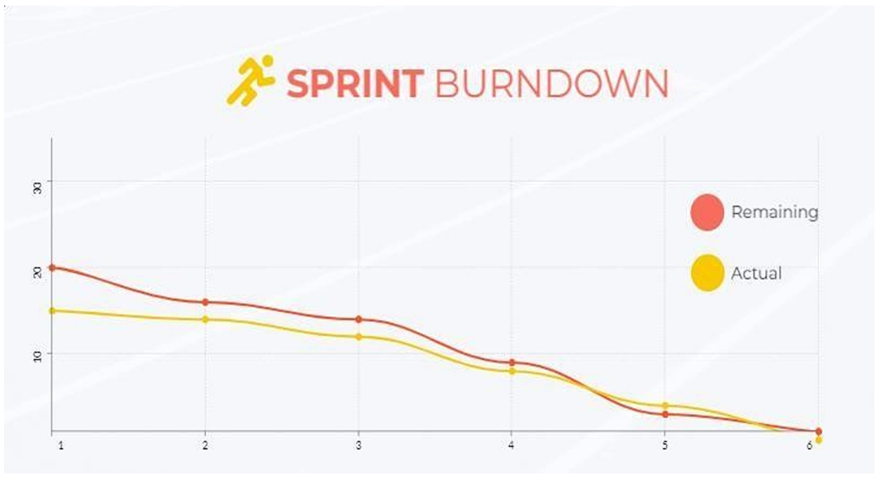
$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

We have a 6-day sprint duration, and the velocity of the team is 20 (points per sprint). So our team's average velocity (AV) per iteration unit (story points per day)

AV = (Sprint Duration / Velocity) = 20 /6 = 3.33

Burndown Chart:

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



7. CODING & SOLUTIONING

7.1 Feature 1

There are a lot of algorithms and a wide variety of data types for phishing detection in the academic literature and commercial products. A phishing URL and the corresponding page have several features which can be differentiated from a malicious URL. For example; an attacker can register long and confusing domain to hide the actual domain name (Cybersquatting, Typosquatting). In some cases attackers can use direct IP addresses instead of using the domain name. This type of event is out of our scope, but it can be used for the same purpose. Attackers can also use short domain names which are irrelevant to legitimate brand names and don't have any FreeUrl addition. But these type of web sites are also out of our scope, because they are more relevant to fraudulent domains instead of phishing domains.

7.2 Feature 2

Beside URL-Based Features, different kinds of features which are used in machine learning algorithms in the detection process of academic studies are used. Features collected from academic studies for the phishing domain detection with machine learning techniques are grouped as given below.

- URL-Based Features
- Domain-Based Features
- Page-Based Features
- Content-Based Features

URL-Based Features

URL is the first thing to analyse a website to decide whether it is a phishing or not. As we mentioned before, URLs of phishing domains have some distinctive points. Features which are related to these points are obtained when the URL is processed.

Domain-Based Features

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.Some of Page-Based Features are given below.

- Global Pagerank
- Country Pagerank
- Position at the Alexa Top 1 Million Site

Content-Based Features

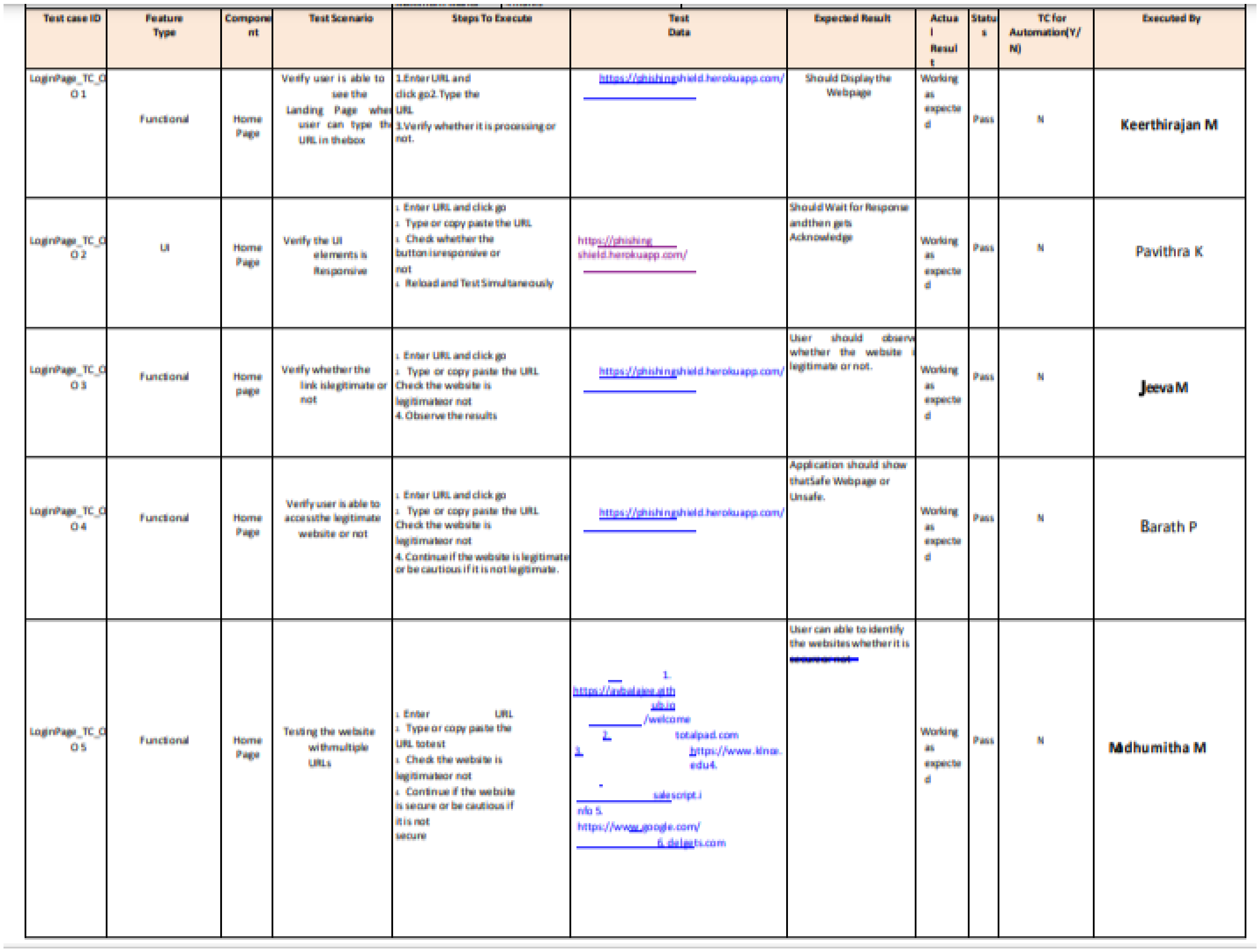
Obtaining these types of features requires active scan to target domain. Page contents are processed for us to detect whether target domain is used for phishing or not. Some processed information about pages are given below.

- Page Titles
- Meta Tags
- Hidden Text
- Text in the Body
- Images etc.

All of features explained above are useful for phishing domain detection. In some cases, it may not be useful to use some of these, so there are some limitations for using these features.

8. TESTING

8.1 Test Cases



Test case ID	Feature Type	Component	Test Scenario	Steps To Execute	Test Data	Expected Result	Actual Result	Status	TC for Automation(Y/N)	Executed By
LoginPage_TC_O O 1	Functional	Home Page	Verify user is able to see the Landing Page when user can type the URL in thebox	1.Enter URL and click go2. Type the URL 3.Verify whether it is processing or not.	https://phishingshield.herokuapp.com/	Should Display the Webpage	Working as expected	Pass	N	Keerthirajan M
LoginPage_TC_O O 2	UI	Home Page	Verify the UI elements is Responsive	1. Enter URL and click go 2. Type or copy paste the URL 3. Check whether the button isresponsive or not 4. Reload and Test Simultaneously	https://phishing shield.herokuapp.com/	Should Wait for Response andthen gets Acknowledge	Working as expected	Pass	N	Pavithra K
LoginPage_TC_O O 3	Functional	Home page	Verify whether the link islegitimate or not	1. Enter URL and click go 2. Type or copy paste the URL Check the website is legitimateor not 4. Observe the results	https://phishingshield.herokuapp.com/	User should observe whether the website is legitimate or not.	Working as expected	Pass	N	Jeeva M
LoginPage_TC_O O 4	Functional	Home Page	Verify user is able to accessthe legitimate website or not	1. Enter URL and click go 2. Type or copy paste the URL Check the website is legitimateor not 4. Continue if the website is legitimate or be cautious if it is not legitimate.	https://phishingshield.herokuapp.com/	Application should show thatSafe Webpage or Unsafe.	Working as expected	Pass	N	Barath P
LoginPage_TC_O O 5	Functional	Home Page	Testing the website withmultiple URLs	1. Enter URL 2. Type or copy paste the URL totest 3. Check the website is legitimateor 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.kinaxedu4. 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	Mdhumitha M

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).

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	70

2.Test Case Analysis

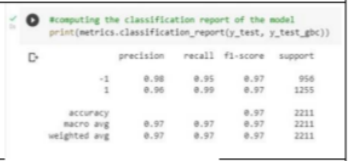
This report shows the number of test cases that have passed, failed, and untested

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

9.1 Performance Metrics

Model Performance Testing: Project team shall fill the following information in model performance testing template.

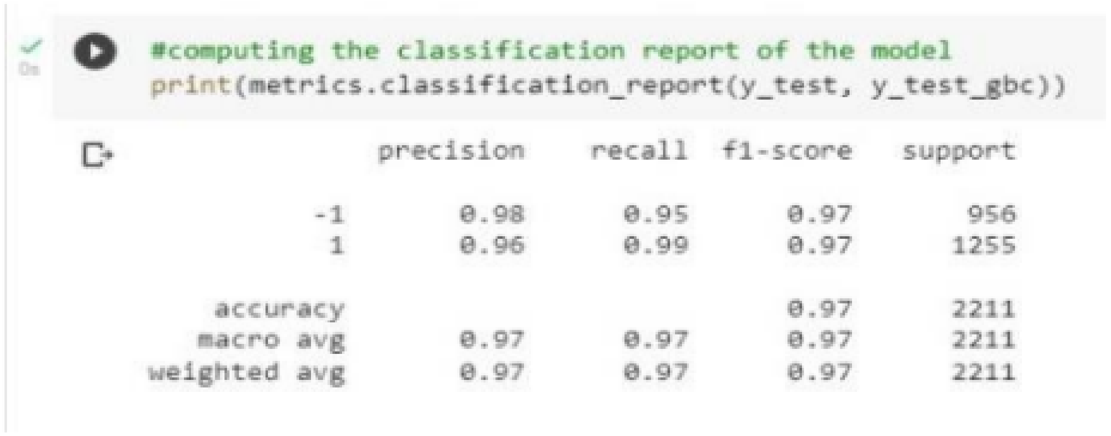
S.No.	Parameter	Values	Screenshot
1.	Metrics	**Classification Model: Gradient Boosting Classification** Accuray Score- 97.1%	
2.	Tune the Model	Hyperparameter Tuning - 97% Validation Method – KFOLD & Cross Validation Method	

```
#computing the classification report of the model
print(metrics.classification_report(y_test, y_test_gbc))

              precision    recall  f1-score   support

          -1       0.98      0.95      0.97       956
           1       0.96      0.99      0.97      1255

    accuracy                           0.97      2211
   macro avg       0.97      0.97      0.97      2211
weighted avg       0.97      0.97      0.97      2211
```

Wilcoxon signed-rank test

```
[illegible]
```

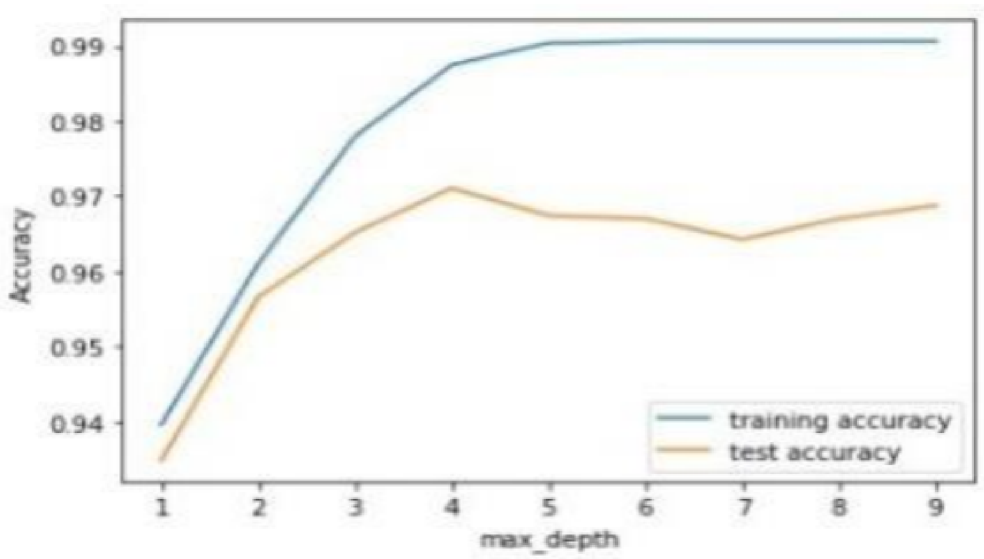
1.METRICS: CLASSIFICATION REPORT:

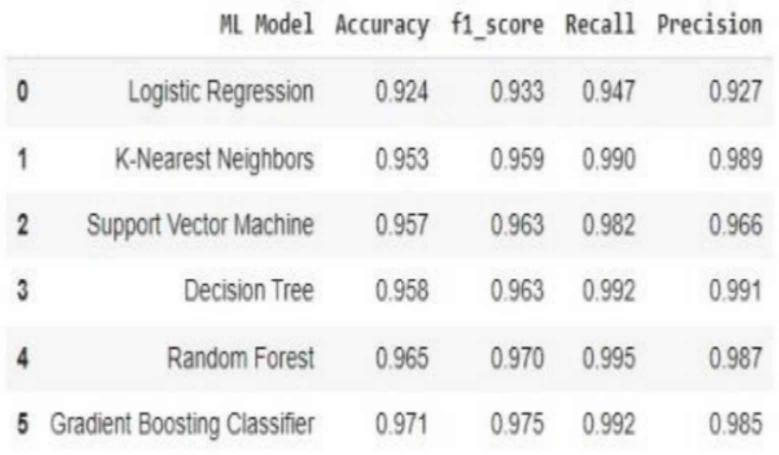


```
#computing the classification report of the model
print(metrics.classification_report(y_test, y_test_gbc))
```

```
              precision    recall  f1-score   support

          -1       0.98      0.95      0.97       956
           1       0.96      0.99      0.97      1255

    accuracy                           0.97      2211
   macro avg       0.97      0.97      0.97      2211
weighted avg       0.97      0.97      0.97      2211
```

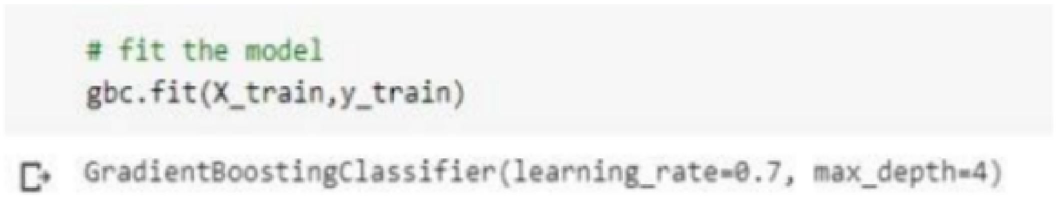
PERFORMANCE :





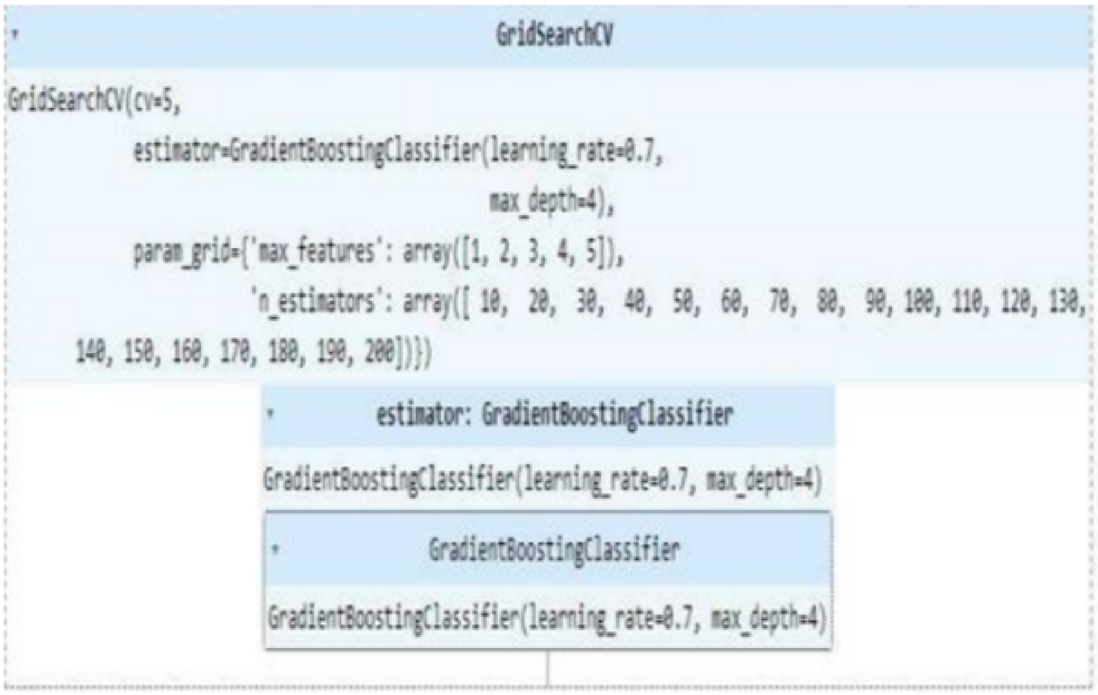
	ML Model	Accuracy	f1_score	Recall	Precision
0	Logistic Regression	0.924	0.933	0.947	0.927
1	K-Nearest Neighbors	0.953	0.959	0.990	0.989
2	Support Vector Machine	0.957	0.963	0.982	0.966
3	Decision Tree	0.958	0.963	0.992	0.991
4	Random Forest	0.965	0.970	0.995	0.987
5	Gradient Boosting Classifier	0.971	0.975	0.992	0.985

1. TUNE THE MODEL – HYPERPARAMETER TUNING



```
# fit the model
gbc.fit(X_train,y_train)
```

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



```
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)
```

10. ADVANTAGES & DISADVANTAGES

ADVANTAGES

- This system can be used by many E-commerce or other websites in order to have good customer relationship.
- User can make online payment securely.
- Data mining algorithm used in this system 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.

DISADVANTAGES

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

11.CONCLUSION

Phishing website attacks are a massive challenge for researchers, and they continue to show a rising trend in recent years. Blacklist/whitelist techniques are the traditional way to alleviate such threats. However, these methods fail to detect non-blacklisted phishing websites (i.e., 0-day attacks). As an improvement, machine learning techniques are being used to increase detection efficiency and reduce the misclassification ratio. However, some of them extract features from third-party services, search engines, website traffic, etc., which are complicated and difficult to access. In this paper, we propose a machine learning-based approach which can speedily and precisely detect phishing websites using URL and HTML features of the given webpage. The proposed approach is a completely client-side solution, and does not rely on any third-party services. It uses URL character sequence features without expert intervention, and hyperlink specific features that determine the relationship between the content and the URL of a webpage. Moreover, our approach extracts TF-IDF character level features from the plaintext and noisy part of the given webpage's HTML.

A new dataset is constructed to measure the performance of the phishing detection approach, and various classification algorithms are employed. Furthermore, the performance of each category of the proposed feature set is also evaluated. According to the empirical and comparison results from the implemented classification algorithms, the XGBoost classifier with integration of all kinds of features provides the best performance. It acquired 1.39% false-positive rate and 96.76% of overall detection accuracy on our dataset. An accuracy of 98.48% with a 2.09% false-positive rate on a benchmark dataset.

12. FUTURE SCOPE

In future work, we plane to include some new features to detect the phishing websites that contain malware. As we said in "Limitations" section, our approach could not detect the attached malware with phishing webpage. Nowadays, blockchain technology is more popular and seems to be a perfect target for phishing attacks like phishing scams on the blockchain. Blockchain is an open and distributed ledger that can effectively register transactions between receiving and sending parties, demonstrably and constantly, making it common among investors. Thus, detecting phishing scams in the blockchain environment is a defiance for more research and evolution. Moreover, detecting phishing attacks in mobile devices is another important topic in this area due to the popularity of smart phones, which has made them a common target of phishing offenses.

13.APPENDIX

13.1 Source Code

1.APP.PY

```
from flask import Flask, render_template, request
import numpy as np
import pandas as pd
from sklearn import metrics
import warnings
import pickle
warnings.filterwarnings('ignore')
from features import FeatureExtraction

app = Flask(__name__)

xgb = pickle.load(open("XGBoostClassifier.pkl", "rb"))

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

        url = request.form["url"]
        obj = FeatureExtraction(url)

        x = np.array(obj.getFeaturesList()).reshape(1,13)
        print(x)
        y_pred =xgb.predict(x)[0]
        print(y_pred)
        y_pro_phishing = xgb.predict_proba(x)[0,0]
        print(y_pro_phishing)
        y_pro_non_phishing = xgb.predict_proba(x)[0,1]
        print(y_pro_non_phishing)

        if(y_pro_phishing*100<60):
            msg="Treat! They say, 'Not all those who wander are lost'.
And you are definitely not lost. Have a safe day exploring!!"
            flag=1
        else:
            msg="Trick! They say, 'Not all those who wander are lost'.
But you are definitely lost. Find other sites to explore!!"
            flag=-1

        return render_template('result.html', msg=msg, url=url,
val=flag)

    return render_template("index.html")

@app.route("/report")
def report():
    return render_template("contact.html")
```

```
if __name__ == '__main__':
    app.run(debug=True)
```

FEATURE.PY

```
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|tin
yurl|tr\.im|is\.gd|cli\.gs|'

'yfrog\.com|migre\.me|ff\.im|tiny\.cc|url4\.eu|twit\.ac|su\.pr|twurl\.n
l|snipurl\.com|'

'short\.to|BudURL\.com|ping\.fm|post\.ly|Just\.as|bkite\.com|snipr\.co
m|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

    # 6.prefixSuffix
    def prefixSuffix(self):
        try:
            match = re.findall('\-', self.domain)
            if match:
                return -1
            return 1
        except:
            return -1

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

    # 8.HTTPS
    def Hppts(self):
        try:
            https = self.urlparse.scheme
            if 'https' in https:
                return 1
            return -1
        except:
            return 1

    # 9.DomainRegLen
    def DomainRegLen(self):
        try:
            expiration_date = self.whois_response.expiration_date
            creation_date = self.whois_response.creation_date
            try:
                if(len(expiration_date)):
                    expiration_date = expiration_date[0]
            except:
                pass
            try:
                if(len(creation_date)):
                    creation_date = creation_date[0]
            except:
                pass

            age = (expiration_date.year-creation_date.year)*12+
(expiration_date.month-creation_date.month)
            if age >=12:
                return 1
            return -1
        except:
            return -1

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

    # 11. NonStdPort
    def NonStdPort(self):
        try:
            port = self.domain.split(":")
            if len(port)>1:
                return -1
            return 1
        except:
            return -1

    # 12. HTTPSDomainURL
    def HTTPSDomainURL(self):
        try:
            if 'https' in self.domain:
                return -1
            return 1
        except:
            return -1

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

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

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

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

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

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

            try:
                percentage = unsafe / float(i) * 100
                if percentage < 31.0:
                    return 1
                elif ((percentage >= 31.0) and (percentage < 67.0)):
                    return 0
                else:
                    return -1
            except:
                return -1

        except:
            return -1

    # 15. LinksInScriptTags
    def LinksInScriptTags(self):
        try:
            i,success = 0,0

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

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

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

    # 16. ServerFormHandler
    def ServerFormHandler(self):
        try:
            if len(self.soup.find_all('form', action=True))==0:
                return 1
            else :
                for form in self.soup.find_all('form', action=True):
                    if form['action'] == "" or form['action'] ==
"about:blank":
                        return -1
                    elif self.url not in form['action'] and
self.domain not in form['action']:
                        return 0
                    else:
                        return 1
        except:
            return -1

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

    # 18. AbnormalURL
    def AbnormalURL(self):
        try:
            if self.response.text == self.whois_response:
                return 1
            else:
                return -1
        except:
            return -1

    # 19. WebsiteForwarding
    def WebsiteForwarding(self):
        try:
            if len(self.response.history) <= 1:
                return 1
            elif len(self.response.history) <= 4:
                return 0
            else:
                return -1
        except:
             return -1

    # 20. StatusBarCust
    def StatusBarCust(self):
        try:
            if re.findall("<script>.+onmouseover.+</script>",
self.response.text):
                return 1
            else:
                return -1
        except:
             return -1

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

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

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

    # 24. AgeofDomain
    def AgeofDomain(self):
        try:
            creation_date = self.whois_response.creation_date
            try:
                if(len(creation_date)):
                    creation_date = creation_date[0]
            except:
                pass

            today  = date.today()
            age = (today.year-creation_date.year)*12+(today.month-
creation_date.month)
            if age >=6:
                return 1
            return -1
        except:
            return -1

    # 25. DNSRecording
    def DNSRecording(self):
        try:
            creation_date = self.whois_response.creation_date
            try:
                if(len(creation_date)):
                    creation_date = creation_date[0]
            except:
                pass

            today  = date.today()
            age = (today.year-creation_date.year)*12+(today.month-
creation_date.month)
            if age >=6:
                return 1
            return -1
        except:
            return -1

    # 26. WebsiteTraffic
    def WebsiteTraffic(self):
        try:
            rank =
BeautifulSoup(urllib.request.urlopen("http://data.alexa.com/data?cli=1
0&dat=s&url=" + url).read(), "xml").find("REACH")['RANK']
            if (int(rank) < 100000):
                return 1
            return 0
        except :
            return -1

    # 27. PageRank
    def PageRank(self):
        try:
            prank_checker_response =
requests.post("https://www.checkpagerank.net/index.php", {"name":
self.domain})

            global_rank = int(re.findall(r"Global Rank: ([0-9]+)",
rank_checker_response.text)[0])
            if global_rank > 0 and global_rank < 100000:
                return 1
            return -1
        except:
            return -1


    # 28. GoogleIndex
    def GoogleIndex(self):
        try:
            site = search(self.url, 5)
            if site:
                return 1
            else:
                return -1
        except:
            return 1

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

    # 30. StatsReport
    def StatsReport(self):
        try:
            url_match = re.search(

'at\.ua|usa\.cc|baltazarpresentes\.com\.br|pe\.hu|esy\.es|hol\.es|swed
dy\.com|myjino\.ru|96\.lt|ow\.ly', url)
            ip_address = socket.gethostbyname(self.domain)
            ip_match =
re.search('146\.112\.61\.108|213\.174\.157\.151|121\.50\.168\.88|192\.
185\.217\.116|78\.46\.211\.158|181\.174\.165\.13|46\.242\.145\.103|121
\.50\.168\.40|83\.125\.22\.219|46\.242\.145\.98|'

'107\.151\.148\.44|107\.151\.148\.107|64\.70\.19\.203|199\.184\.144\.2
7|107\.151\.148\.108|107\.151\.148\.109|119\.28\.52\.61|54\.83\.43\.69
|52\.69\.166\.231|216\.58\.192\.225|'
'118\.184\.25\.86|67\.208\.74\.71|23\.253\.126\.58|104\.239\.157\.210|
175\.126\.123\.219|141\.8\.224\.221|10\.10\.10\.10|43\.229\.108\.32|10
3\.232\.215\.140|69\.172\.201\.153|'

'216\.218\.185\.162|54\.225\.104\.146|103\.243\.24\.98|199\.59\.243\.1
20|31\.170\.160\.61|213\.19\.128\.77|62\.113\.226\.131|208\.100\.26\.2
34|195\.16\.127\.102|195\.16\.127\.157|'

'34\.196\.13\.28|103\.224\.212\.222|172\.217\.4\.225|54\.72\.9\.51|192
\.64\.147\.141|198\.200\.56\.183|23\.253\.164\.103|52\.48\.191\.26|52\
.214\.197\.72|87\.98\.255\.18|209\.99\.17\.27|'

'216\.38\.62\.18|104\.130\.124\.96|47\.89\.58\.141|78\.46\.211\.158|54
\.86\.225\.156|54\.82\.156\.19|37\.157\.192\.102|204\.11\.56\.48|110\.
34\.231\.42', ip_address)
            if url_match:
                return -1
            elif ip_match:
                return -1
            return 1
        except:
            return 1

    def getFeaturesList(self):
        return self.features
```

INDEX.HTML

```
<!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="VAIBHAV BICHAVE">
    <!-- 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 URL DETECTION</h2>

            <br>
            <form action="/" method ="post">
                <input type="text" class="form__input" name ='url'
id="url" placeholder="Enter URL" required="" />
                <label for="url" class="form__label">URL</label>
                <button class="button" role="button" >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.m
in.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>

</body>

</html
```

13.2 GitHub & Project Demo Link

GITHUB Link:https://github.com/IBM-EPBL/IBM-Project-10260-1668680963

Demo Link:https://www.canva.com/design/DAFSW_qxfL0/b1vOKrIrP6C-yvi4PF-Sxg/view?utm_content=DAFSW_qxfL0&utm_campaign=designshare&utm_medium=link&utm_source=recording_view