M.KUMARASAMY
COLLEGE OF ENGINEERING
NAAC Accredited Autonomous Institution
Approved by AICTE & Affiliated to Anna University
ISO 9001:2015 & ISO 14001:2015 Certified Institution
Thalavapalayam, Karur – 639 113.

A Project Report

on

WEB PHISHING DETECTION

Submitted in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

Under the Guidance of

Mrs. K. MAKANYADEVI M.E.,
Assistant Professor/CSE

Submitted by

TEAM ID: PNT2022TMID15588

927619BCS4120	**K.K.VARSHAA**
927619BCS4119	**S.TRINAYA**
927619BCS4115	**R.SUDHARSAN**
927619BCS4097	**S.RANJITH**

NAALAIYA THIRAN – EXPERIENTIAL PROJECT BASED LEARNING INITIATIVE

18CSE040L - PROFESSIONAL READINESS FOR INNOVATION,EMPLOYABILITY AND ENTERPRENURSHIP

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

M.KUMARASAMY COLLEGE OF ENGINEERING

(Autonomous)

Karur - 639 113

November, 2022

TABLE OF CONTENT

ABSTRACT

Web Spoofing lures the user to interact with the fake websites rather than the real ones. The main objective of this attack is to steal the sensitive information from the users. The attacker creates a ‘shadow’ website that looks similar to the legitimate website. This fraudulent act allows the attacker to observe and modify any information from the user. This paper proposes 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 Phistank and Yahoo directory datasets. The obtained results show that the detection mechanism is deployable and capable to detect various types of phishing attacks maintaining a low rate of false alarms.

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

Phishing costs Internet users billions of dollars per year. It refers to luring techniques used by identity thieves to fish for personal information in a pond of unsuspecting Internet users. Phishers use spoofed e-mail, phishing software to steal personal information and financial account details such as usernames and passwords. This paper deals with methods for detecting phishing Web sites by analyzing various features of benign and phishing URLs by Machine learning techniques. We discuss the methods used for detection of phishing Web sites based on lexical features, host properties and page importance properties. We consider various machine learning algorithms for evaluation of the features in order to get a better understanding of the structure of URLs that spread phishing. The fine-tuned parameters are useful in selecting the apt machine learning algorithm for separating the phishing sites from benign sites.

1.2 PURPOSE

The main purpose of the project is to detect the fake or phishing websites who are trying to get access to the sensitive data or by creating the fake websites and trying to get access of the user personal credentials. We are using machine learning algorithms to safeguard the sensitive data and to detect the phishing websites who are trying to gain access on sensitive data.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.

CHAPTER 2

LITERATURE SURVEY

2.1 EXISITING PROBLEM

Phishing Detection techniques do suffer low detection accuracy and high false alarm especially when novel phishing approaches are introduced. Besides, the most common technique used, blacklist-based method is inefficient in responding to emanating phishing attacks since registering new domain has become easier, no comprehensive blacklist can ensure a perfect up-to-date database. 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 websitedetection with varying degrees of accuracy. Therefore, ensemble can be seen to be a better solution as it can combine the similarity in accuracy and different error-detection rate properties in selected algorithms.

2.2 REFERENCES

1. Adebowale, M. A., Lwin, K. T., Sánchez, E. and Hossain, M. A. (2019) 'Intelligent web-phishing detection and protection scheme using integrated features of Images, frames and text', Expert Systems with Applications, 115, pp. 300-313.
2. Aggarwal, A., Rajadesingan, A. and Kumaraguru, P. 'PhishAri: Automatic real-time phishing detection on twitter', eCrime Researchers Summit (eCrime), Las Croabas, Puerto Rico, 23-24 Oct. 2012: IEEE, 1- 12.
3. Lee, J.-L. and Park, D.-h. (2016) 'Phishing Detection Using Web Site Heuristics', International Information Institute (Tokyo), 19(2), pp. 523-530.
4. Yang, P., Zhao, G. and Zeng, P. (2019) 'Phishing Website Detection Based on

Multidimensional Features Driven by Deep Learning', IEEE Access, 7, pp. 15196-15209.

5. Mohammad R., Thabtah F. McCluskey L., (2015) Phishing websites dataset. Available: https://archive.ics.uci.edu/ml/datasets/Phishing+Websites Accessed January 2016.
6. Gunter Ollmann, "The Phishing Guide Understanding & Preventing Phishing Attacks", IBMInternet Security Systems, 2007.
7. http://dataaspirant.com/2017/01/30/how-decision-tree-algorithm-works/
8. https://www.kdnuggets.com/2016/07/support-vector-machines-simple-explanation.html
9. https://resources.infosecinstitute.com/category/enterprise/phishing/thephishing-landscape/phishing-data-attack-statistics/#gref
10. http://dataaspirant.com/2017/05/22/random-forest-algorithm-machine-learing/

2.3 PROBLEM STATEMENT DEFINITION

Phishing is one of the techniques which are used by the intruders to get access to the user credentials or to gain access to the sensitive data. This type of accessing the is done by creating the replica of the websites which looks same as the original websites which we use on our daily basis but when a user click on the link he will see the website and think its original and try to provide his credentials . To overcome this problem we are using some of the machine learning algorithms in which it will help us to identify the phishing websites based on the features present in the algorithm. By using these algorithm we cam be able to keep the user personal credentials or the sensitive data safe from the intruders.

CHAPTER 3

IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

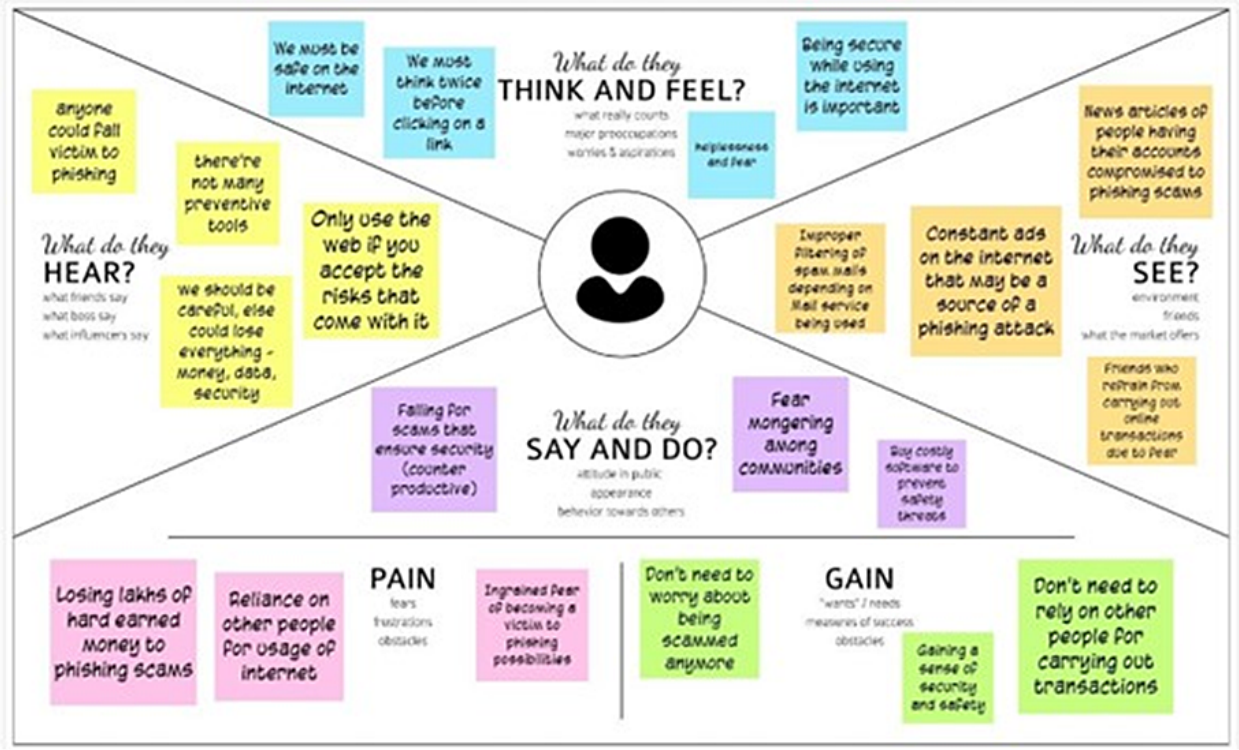

Figure 3.1 Empathy Map

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.

3.2 IDEATION AND BRAINSTORMING

Step-1: Team Gathering, Collaboration and Select the Problem Statement

A principal difference between ideation and brainstorming is that ideation is commonly more thought of as being an individual pursuit, while brainstorming is almost always a group activity.

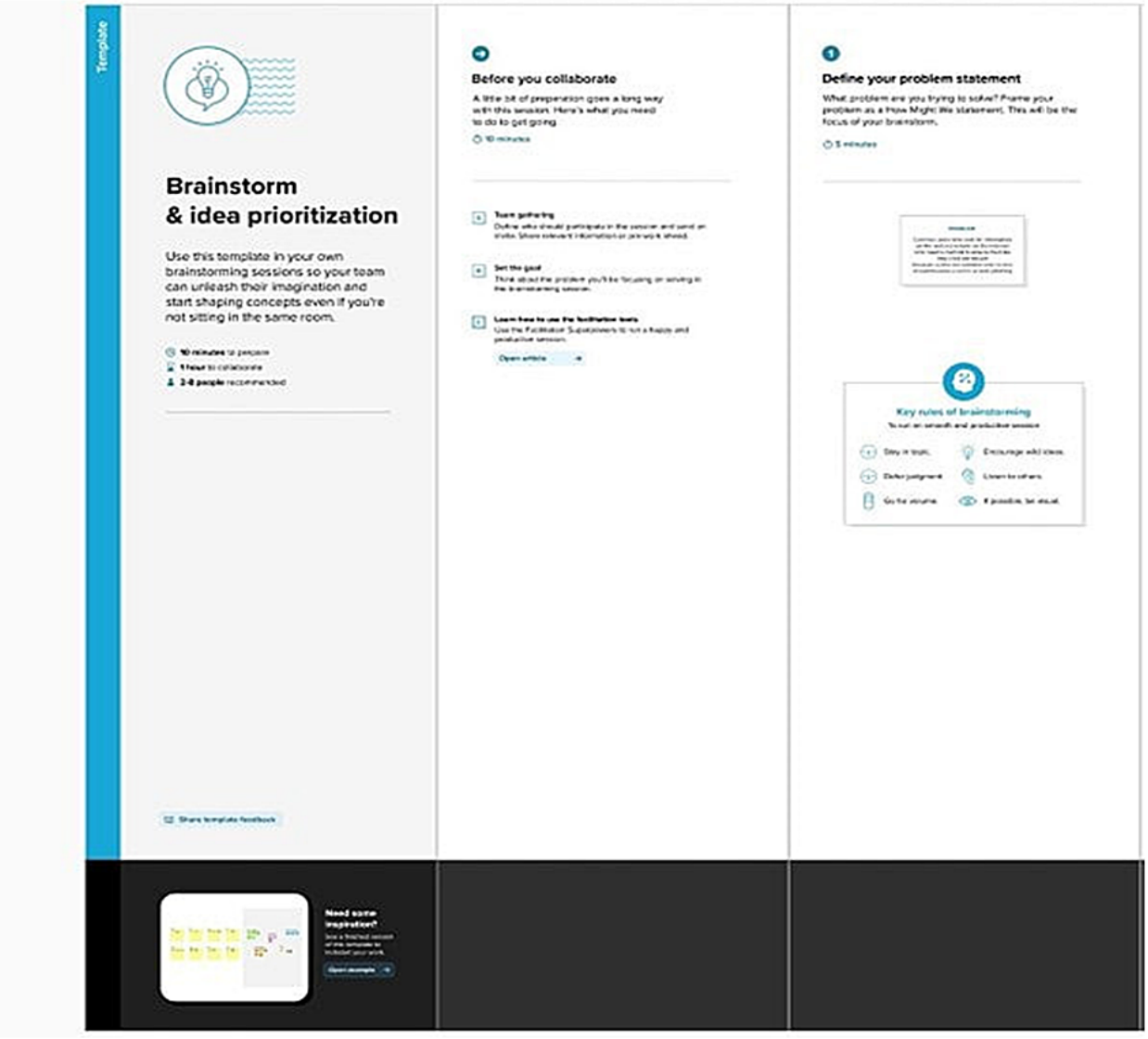

Figure 3.2 Ideation and Brainstorming

Step-2: Brainstorm, Idea Listing and Grouping

The idea listing and grouping is used to organize and analyse large numbers of ideas by categorising them. By organising and reorganising ideas, students gain a better appreciation of, and dialogue about, their ideas. As students create idea clusters, new contexts and connections among themes emerge.

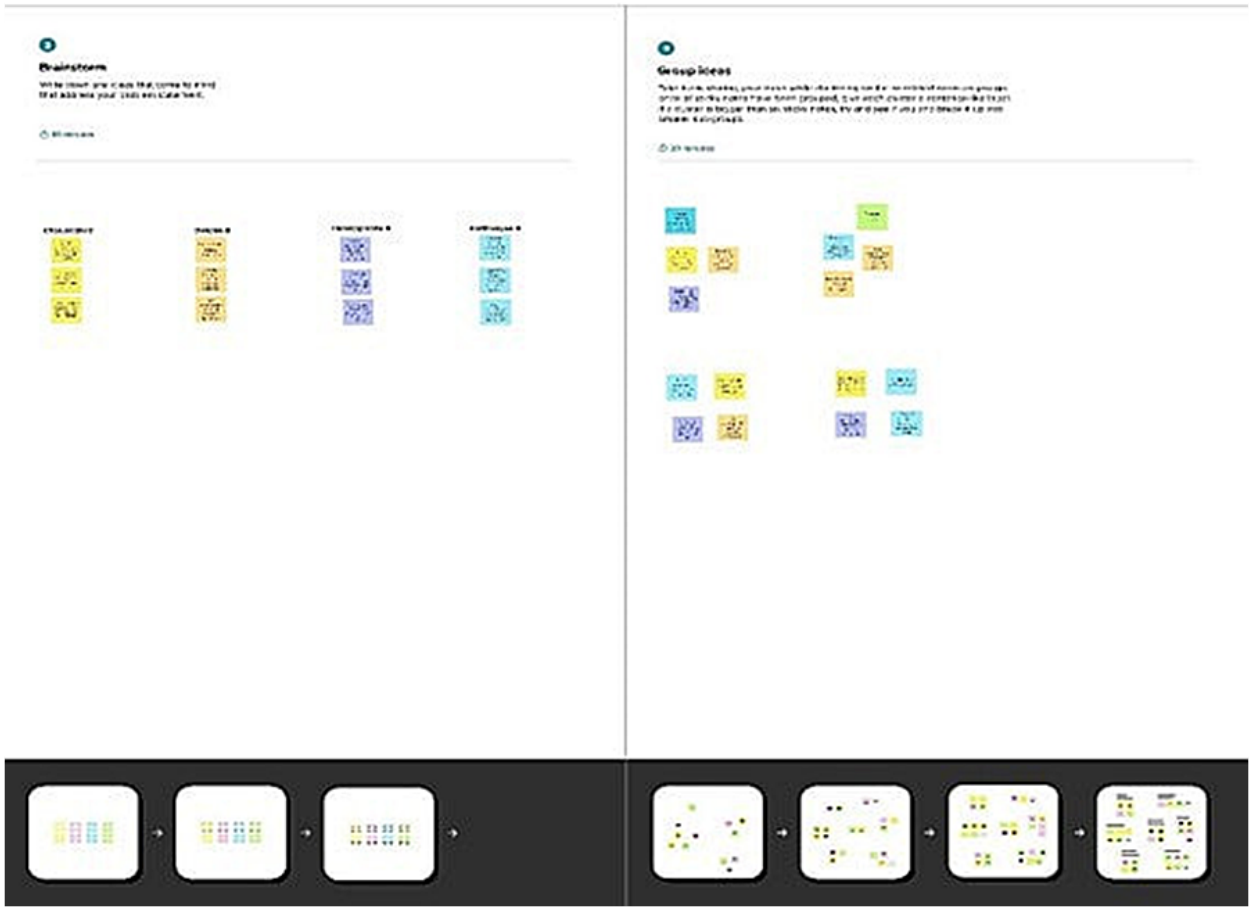

Figure 3.3 Ideation and Brainstorming

Step-3 : Idea Prioritization

Idea prioritization is just a part of the idea management process. Having a structured idea management process and a systematic way of gathering, evaluating and prioritizing new ideas takes time. To make it work, the entire idea management process should be integrated to the everyday ways of working.

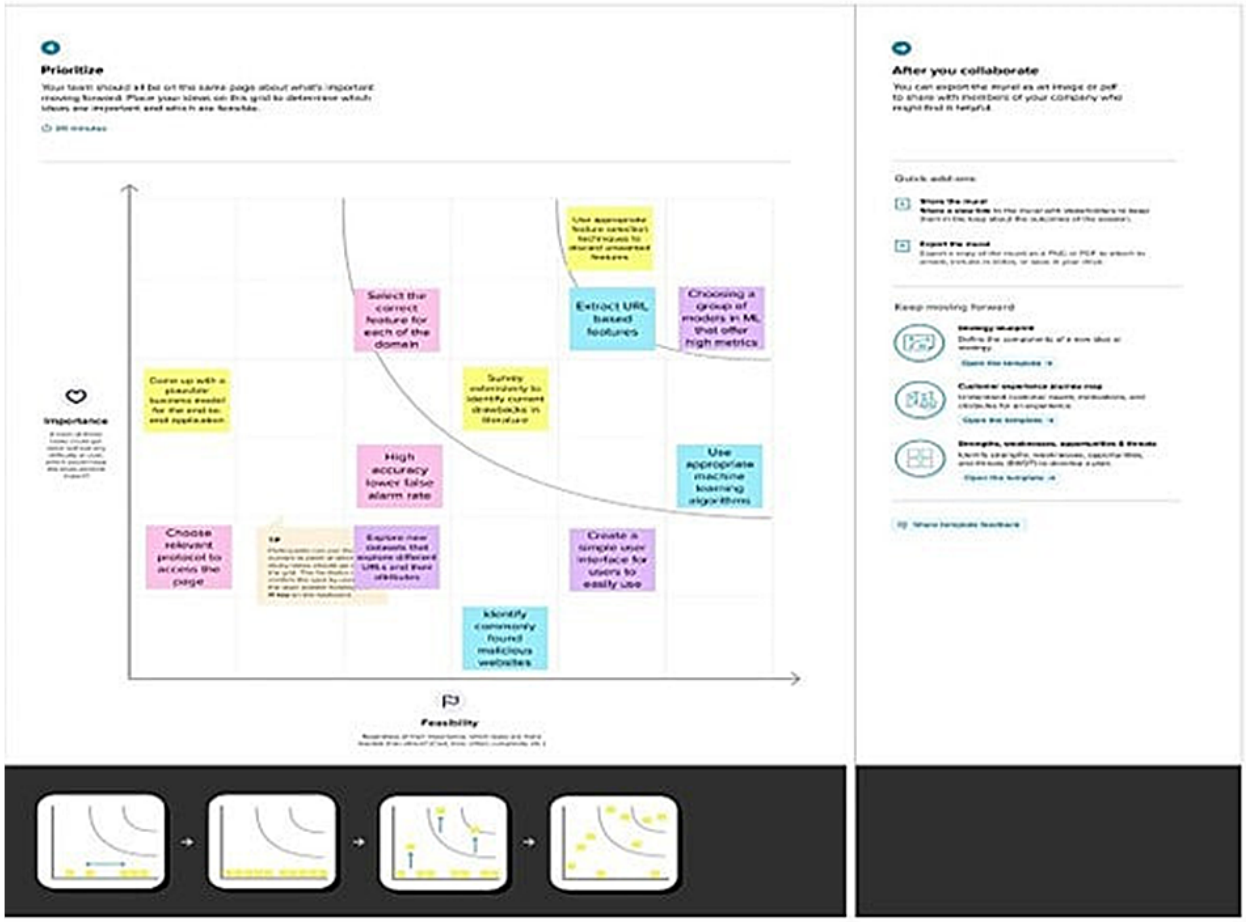

Figure 3.4 Ideation and Brainstorming

3.3 PROPOSED SOLUTION

S. No.	Parameter	Description
1.	**Problem Statement(Problemtobesolved)**	To reduce the people falling for web phishing scams by creating a sophisticated tool that classifies a website as maliciousor safe to use
2.	**Idea/Solutiondescription**	Identify web phishing,classify whether it is an attack and prevent malicious intrusive websites
3.	**Novelty/Uniqueness**	1. Uses an Ensemble model 2. Explores weighted features for Neural Network approaches 3. Extensive feature extraction strategy from the URL 4. Simple,Easy-to-UnderstandUI
4.	**Social Impact /CustomerSatisfaction**	1. Users need not fear of losing lakhs of hard earned money to phishings cams & Users need not feels cared to use the internet 2. Primarily targets the benefit of senior citizens and technologically challenged sections of the society 3. Customers don't need to rely on offline transactions because of the fear of initiating transactions online
5.	**Business Model(RevenueModel)**	1. B2B (Machine Learning model/API can be sold to various companies for their employees) and B2C Model (End product sold to individuals such as children's devices and senior citizens prone to attacks) 2. Site can charge a one time fee for a device/user based on demographic surveys(Rs.50peryear) 3. Companies can be charged a discounted fee due to bulk purchase of the Application Programming Interface(API) 4. Premium users will have access to details of the URL and reasonings for why a site has been classified 'unsafe'.

Table 3.5 Proposed Solution

3.4 PROBLEM SOLUTION FIT

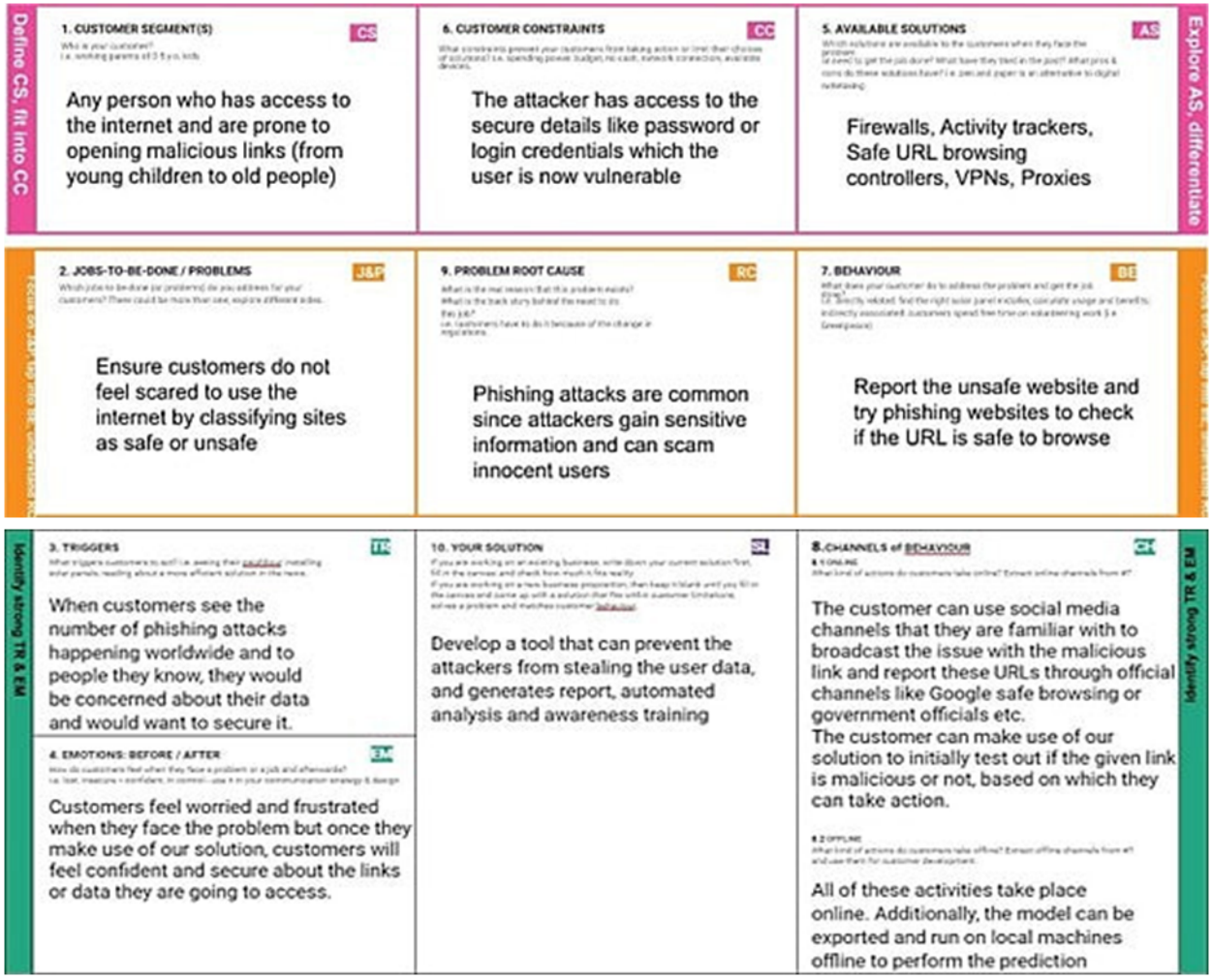

Figure 3.6 Problem Solution Fit

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

Functional requirements are product features or functions that developers must implement to enable users to accomplish their tasks.

FR NO.	FunctionalRequirement (Epic)	SubRequirement(Story/Sub-Task)
FR-1	UserInput	UserinputsanURLinrequiredfield tocheck itsvalidation.
FR-2	WebsiteComparison	Model compares the websites using Blacklist and White list approach
FR-3	Featureextraction	After comparing, if none found on comparison then it extracts feature using heuristic and visual similarity approach.
FR-4	Prediction	Model predicts the URL using Machine Learning algorithms such as LogisticRegression ,KNN
FR-5	Classifier	Model sends all output to classifier and produces final result.

FR-6	Announcement	Model then displays whether website is alegal site or a phishing site.
FR-7	Events	This model needs the capability of retrieving and displaying accurate result for a website.

Table 4.1 Functional Requirements

4.2 NON-FUNCTIONAL REQUIREMENTS

Non functional Requirements (NFRs) define system attributes such as security, reliability, performance, maintainability, scalability, and usability. They serve as constraints or restrictions on the design of the system across the different backlogs.

FR No.	**Non-Functional Requirement**	**Description**
NF R-1	Usability	Analysis of consumers' product usability in the design process with user experience as the core may certainly help designers better grasp users' prospective demands in webphishing detection,behaviour,and experience.
NF R-2	Security	It guarantees that any data included within the systemor its components will be safe from malware threats or unauthorised access.If you wish to prevent unauthorised access to the admin panel, describe the login flow and different user roles as system behaviouror user actions.
NF R-3	Reliability	It specifies the likelihood that the system or its component will operate without failure for aspecific amount oftime under prescribed conditions.
NF R-4	Performance	It is concerned with a measurement of the system's reaction time under various load circumstances.

NF R-5	Availability	It represents the likelihood that a user will beable to access the system at a certain moment in time.While it can be represented as an expected proportion of successful requests, it can also be defined as a percentage of time the system is operational within a certain time period.

Table 4.2 Non Functional Requirements

CHAPTER 5

PROJECT DESIGN

5.1 DATAFLOW DIAGRAM

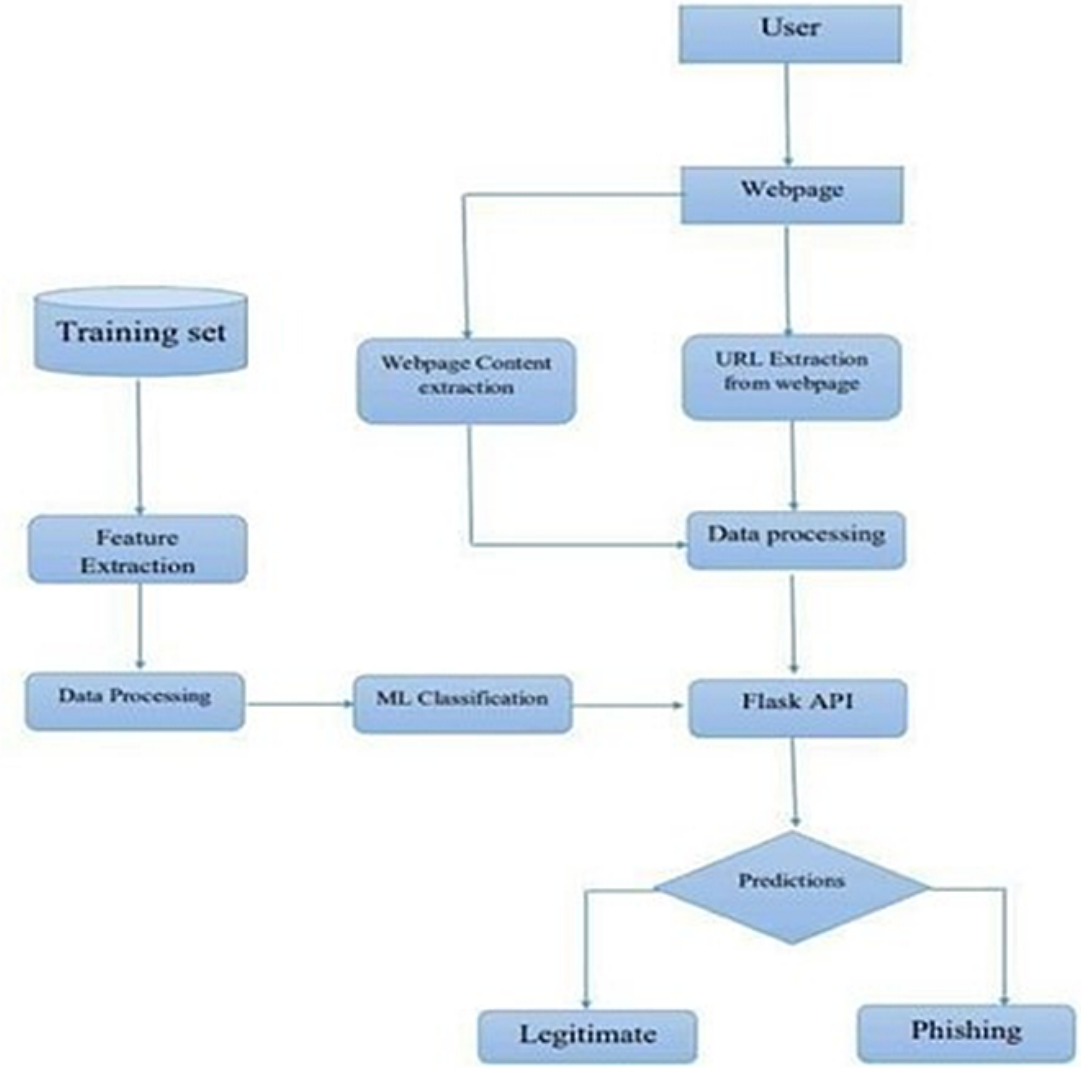

Figure 5.1 Dataflow Diagram of Web Phishing Detection

1. Website data is collected from Kaggle
2. Exploratory data analysis done on the input dataset.
3. Then removal of null values, duplicates and outliers.
4. Then the dependent and independent variable is defined.
5. Train test split is done.
6. Regression model is built.
7. Then the model is fitted with front end which is developed using HTML, CSS with the help of Python Flask Web Framework.
8. Finally, the output will be predicted for the user input.

5.2 SOLUTION AND TECHNICAL ARCHITECTURE

The solution architecture which can explains to collect the dataset and move to pre-processing, then need to train and test datas with the help of machine learning algorithms, to predict the output. Technical architecture includes the major components of the system, their relationships, and the contracts that define the interactions between the components. The goal of technical architects is to achieve all the business needs with an application that is optimized for both performance and security.

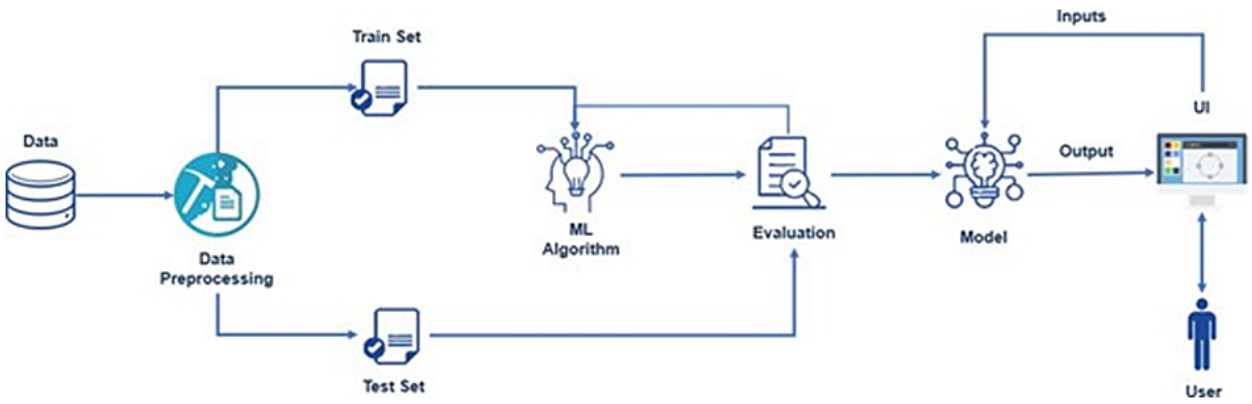

Figure 5.2 Solution and technical architecture

5.3 USER STORY

User Type	Functional Requirement (Epic)	User Story number	User Story/Task	Acceptance Criteria	Priorty	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application before entering my email,password, and cofirming my password	I can access my account /dashboard	High	Sprint-1
		USN-2	As a user,I will receive confirmation email once I have registerd for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user ,Ican 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		Medium	Sprint-1

	Login	USN-5	As a user,I can log into the application by entering email & password		High	Sprint-1
Customer (Web User)	User Input	USN-1	As a user I can input the particular URL in the required field and waiting for validation	I can go access the website without any problem	High	Sprint-1
Customer Care Executive	Feature Extraction	USN-1	After I compare in case if none found on comparison then we can extract feature	As a User I can have comparison between websites for security	High	Sprint-1
Administrator	Prediction	USN-1	Here the Model will predict the URL websites using Machine Learning algorithm such as Logistic Regression	In this I can have correct prediction on the particular algorithms	High	Sprint-1

	Classifier	USN-2	Here I will send all the model output to classifier in order to produce final result	In this I will find the correct classifier for producing the result	Medi um	Sprint-2

CHAPTER 6

PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

Sprint	Functional Requirement(Epic)	User StoryNumber	UserStory/Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	Collect the appropriated at a set for detecting phishing website.	10	High	Sudharsan R
Sprint-1	DataPreprocessing	USN-2	Used to transform the data into useful format.	8	Medium	Sudharsan R
Sprint-2	Model Building	USN-3	Detect the phishing Website.	10	High	Trinaya S
Sprint-2	Model Building	USN-4	Splitting the Model intoTraining and Testing from the overall dataset	8	Medium	Trinaya S
Sprint-3	Training andTesting	USN-5	Train the Model using Logistic Regression algorithm and Testing the Performance of the model.	10	High	Varshaa K K
Sprint-3	Application Building	USN-6	Build the HTML and Python code	8	Medium	Varshaa K K

Sprint-4	Implementation of the Application	USN-7	RunFlask App	10	High	Ranjith S
Sprint-4	Implementation of the Application	USN-8	Deploy the Model on IBM Cloud.	8	Medium	Ranjith S

Table 6.1 Sprint Planning and Estimation

6.2 PROJECT TRACKER, VELOCITY & BURNDOWN CHART

Sprint	Total Story Points	Duration	Sprint StartDate	Sprint EndDate(Planned)	Story Points Completed (as on Planned EndDate)	Sprint Release Date (Actual)
Sprint-1	10	6Days	24Oct2022	29Oct2022	8	29Oct2022
Sprint-2	10	6Days	31Oct2022	05 Nov 2022	7	05 Nov 2022
Sprint-3	10	6Days	07 Nov 2022	12 Nov 2022	8	12 Nov 2022
Sprint-4	10	6Days	14 Nov 2022	19 Nov 2022	7	19 Nov 2022

Table 6.2 Project Tracker,Velocity&Burndown Chart

VELOCITY

Imagine we have 6 – days sprint duration, and the velocity of the team is 10 (points persprint).Let's calculatethe team'saverage(AV)per iterationunit (Storypointsperday).

AV = Sprint Duration/Velocity

= 6/10

= 0.6

6.3 PROJECT DELIVERY SCHEDULE

TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Literature survey on the selected project & gathering information by referring the technical papers, research publications,journals etc.	8SEPTEMBER 2022
Prepare Empathy Map	Prepare Empathy Map Canva sto capture the user Pains &Gains, prepare list of problem statements that are to be solved by this project.	8SEPTEMBER 2022
Ideation	List the ideas by organizing a brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	16SEPTEMBER 2022

Proposed Solution	Prepare the proposed solution document, which includes novelty, feasibility of idea,revenue model, social impact,scalability of solution,etc.	23SEPTEMBER 2022
Solution Architecture	Prepare Solution Architecture document.	27SEPTEMBER 2022
Problem Solution Fit	Prepare problem-solution fit document.	1OCTOBER 2022

CustomerJourney	Prepare the customer journey maps to understand the user interactions & experiences with the application (entry to exit).	6OCTOBER 2022
Functional Requirement	Prepare the functional requirement document.	15OCTOBER 2022
Data Flow Diagrams	Draw the data flow diagrams and submit for review.	15OCTOBER 2022
Technology Architecture	Prepare the technology architecture diagram.	16OCTOBER 2022
Prepare Milestone & Activity List	Prepare the milestones & activity list of the project.	21OCTOBER 2022
Project Development - Delivery of Sprint-1,2,3&4	Develop & submit the developed code by testing it.	19NOVEMBER 2022

Table 6.3 Project Delivery Schedule

CHAPTER 7

CODING AND SOLUTION

7.1 FEATURE 1

Decision Tree Classifier is used to train and test the model for detecting the phishing website with the help of collected and preprocessed dataset collections. NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. Moreover, NumPy forms the foundation of the Machine Learning stack. Pandas is an open-source Python package that is most widely used for data science/data analysis and machine learning tasks. Sea born is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. For a brief introduction to the ideas behind the library, you can read the introductory notes or the paper.

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible. Create publication quality plots. Make interactive figures that can zoom, pan, update.EDA is applied to investigate the data and summarize the key insights. It will give you the basic understanding of your data, it is distribution, null values and much more. You can either explore data using graphs or through some python functions. There will be two types of analysis. Descriptive statistics are brief informational coefficients that summarize a given data set, which can be either a representation of the entire population or a sample of a population. Descriptive statistics are broken down into measures of central tendency and measures of variability.

Measures of central tendency include the mean, median, and mode, while measures of

variability include standard deviation, variance, minimum and maximum variables, kurtosis, and Skewness. Label Encoding refers to converting the labels into a numeric form to convert them into the machine-readable form. Machine learning algorithms can then decide in a better way how those labels must be operated. It is an important pre-processing step for the structured dataset in supervised learning. “Pickling” is the process whereby a Python object hierarchy is converted into a byte stream, and “unpickling” is the inverse operation, whereby a byte stream is converted back into an object hierarchy. 19 XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible, and portable. It implements machine learning algorithms under the Gradient Boosting framework.

7.2 FEATURE 2

The framework is the basis upon which software programs are built. It serves as a foundation for software developers, allowing them to create a variety of applications for certain platforms. It is a set of functions and predefined classes used to connect with the system software and handle inputs and outputs. It simplifies the life of a developer while giving them the ability to use certain extensions and makes the online applications scalable and maintainable. Flask is a web application framework written in Python. A Web Application Framework or a simply a Web Framework represents a collection of libraries and modules that enable web application developers to write applications without worrying about low-level details such as protocol, thread management, among other examples. Flask is a web application framework written in Python.

Flask is based on the Werkzeg WSGI toolkit and the Jinja2 template engine. Both are Pocco projects. The Web Server Gateway Interface (Web Server Gateway Interface, WSGI) has been used as a standard for Python web application development. WSGI is the specification of a common interface between web servers and web applications. Flask is often referred to as a micro-framework. It is designed to keep the core of the application simple and scalable. Instead of an abstraction layer for database support, Flask supports extensions to add such capabilities to the application. Unlike the Django framework, Flask is very Pythonic. It’s easy to get started with Flask, because it doesn’t have a huge learning

curve.HTML stands for Hyper Text Markup Language. HTML is the standard markup language for creating Web pages. HTML describes the structure of a Web page. HTML consists of a series of elements. HTML elements tell the browser how to display the content. Flask is used for developing web applications using python, implemented on Werkzeug and Jinja2. Advantages of using Flask framework are: There is a built-in development server and a fast debugger provided. The model deployed using Flask is used to predict the Chronic Kidney Disease. Hypertext markup language (HTML) is the basic language used to create documents for the Web and, along 20 with HTTP (hypertext transfer protocol) and URLs (universal resource locators), is one of the three main protocols of the Web. Hypertext is text that contains hyperlinks. A hyperlink is an automated cross-reference to another location on the same document or to another document which, when selected by a user, causes the computer to display the linked location or document within a concise period.

A markup language is a set of tags that can be embedded in digital text to provide additional information about it, including its content, structure and appearance. This information facilitates automated operations on the text, including formatting it for display, searching it and even modifying it. Some type of markup language is employed by every word processing program and by nearly every other program that displays text, although such languages and their tags are typically hidden from the user.HTML consists of a set of predefined tags that can be embedded in text by web site designers in order to indicate the details of how web pages are rendered (i.e., converted into a final, easily usable, form) by web browsers. These details include paragraphing, margins, fonts (including style and size), columns, colors (background and text), links, the location of images, text flow around images, tables, and user input form elements (such as spaces for adding text and submit buttons).

CHAPTER 8

TESTING

8.1 TEST CASES

```
In [24]: y_pred_lr=lr.predict(X_test)
         from sklearn.metrics import accuracy_score
         y_test_lr = lr.predict(X_test)
         y_train_lr = lr.predict(X_train)

         acc_train_lr = accuracy_score(y_train,y_train_lr)*100
         acc_test_lr = accuracy_score(y_test,y_test_lr)*100

         storeResults('Logistic Regression', acc_train_lr, acc_test_lr)
         log_reg=accuracy_score(y_test,y_pred_lr)*100
         print("Logistic Regression: Accuracy: {:.3f}".format(log_reg))
         print("Classification Report : ")
         print(classification_report(y_test,y_pred_lr))

Logistic Regression: Accuracy: 92.282
Classification Report :
              precision    recall  f1-score   support

          -1       0.93      0.90      0.91      1498
           1       0.92      0.95      0.93      1819

    accuracy                           0.92      3317
   macro avg       0.92      0.92      0.92      3317
weighted avg       0.92      0.92      0.92      3317
```

Figure 8.1 Execution

8.2 USER ACCEPTANCE TESTING

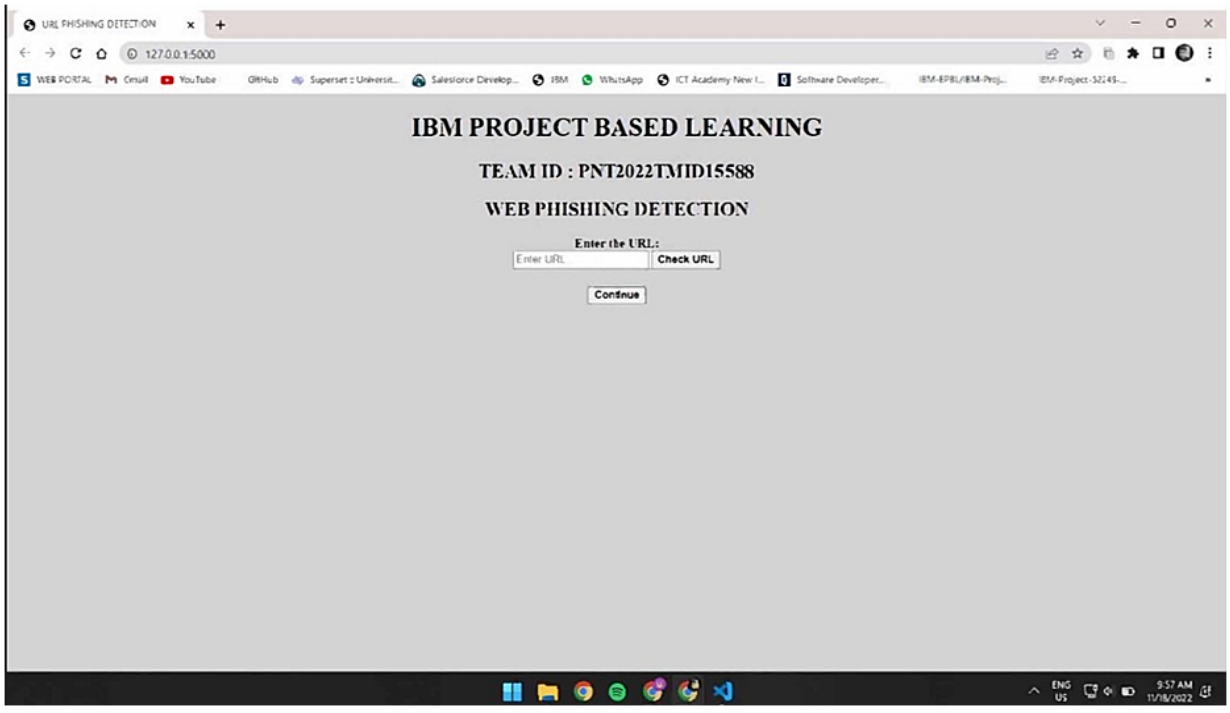

Figure 8.2 Project page

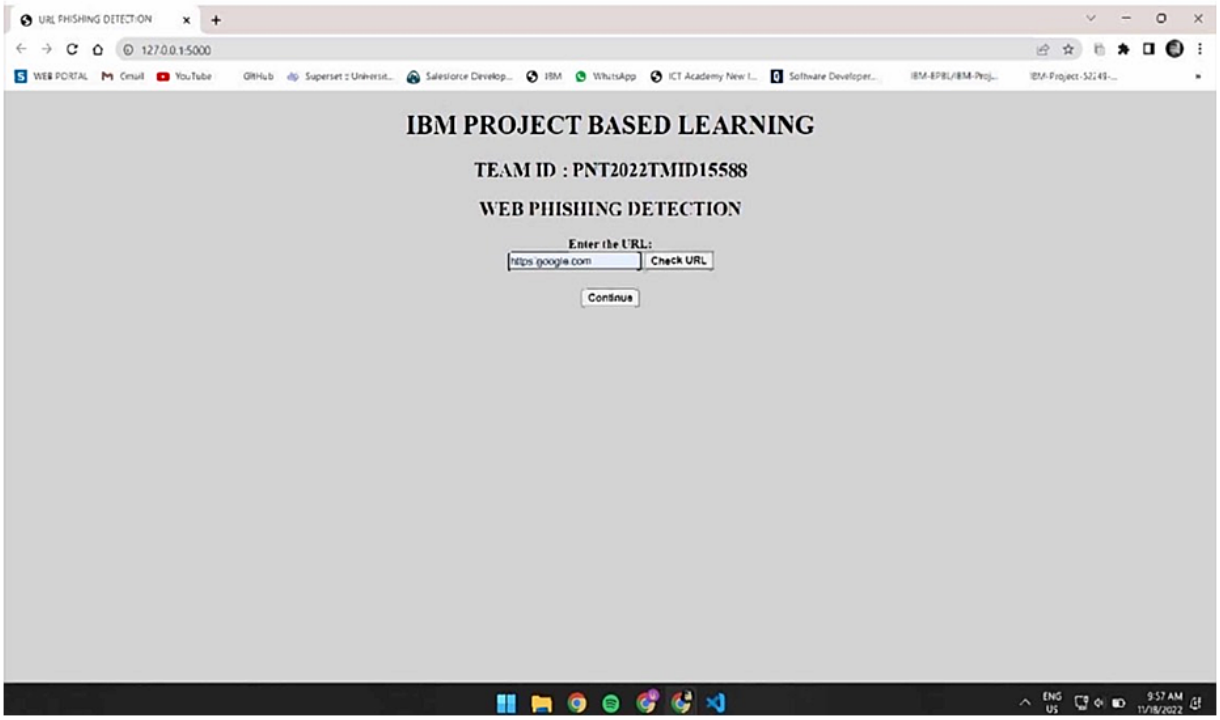

Figure 8.3 Enter the URL

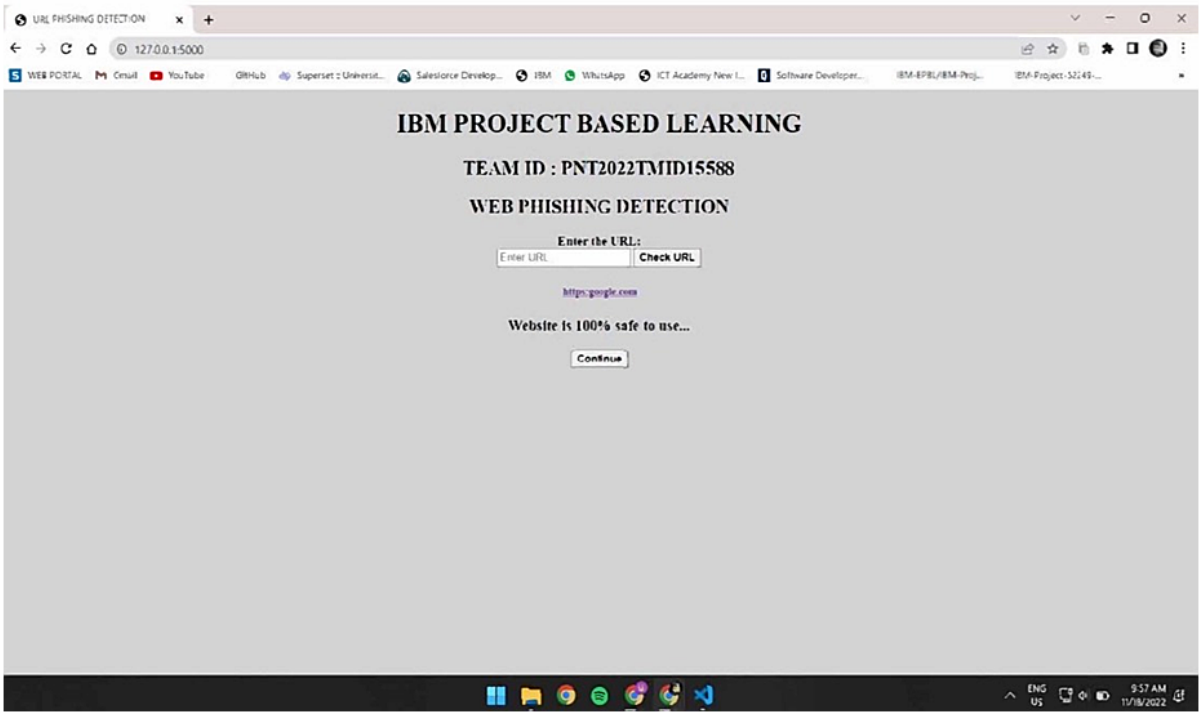

Figure 8.4 Check URL

CHAPTER 9

RESULTS

9.1 PERFORMANCE METRICS

Confusion Matrix: The confusion matrix is used to measure the introduction of two class issue for the given instructive record. The right corner to corner parts TP(True positive) and TN (True Negative) adequately describe instances similarly as FP (false positive) and FN (false negative) wrongly request instances. Confusion Matrix correctly classify instance TP+TN incorrectly classify instances.

- True positives imply the phishing website that were precisely named by the classifier.
- True negatives are the legitimate website that were precisely set apart by the classifier.
- False positives are the legitimate that were erroneously set apart as phishing website
- False negatives are the phishing website that were incorrectly stamped legitimate website.

	Classified as phishing	Classified as legitimate
Phishing website	True Positive (TP)	False Negative (FN)
Legitimate website	False Positive (FP)	True Negative (TN)

Figure 9.1 Performance Metrics

Classification Report: A Classification report is used to measure the quality of predictions from a classification algorithm. How many predictions are True and how many are False. More specifically, True Positives, False Positives, True Negatives and False Negatives are used to predict the metrics of a classification report .

1. Precision is the ability of a classifier not to label an instance positive that is actually negative. For each class it is defined as the ratio of true positives to the sum of true and false positives.

$$\text{Precision} = TP/(TP + FP)$$

1. Recall is the ability of a classifier to find all positive instances. For each class it is defined as the ratio of true positives to the sum of true positives and false negatives.

$$\text{Recall} = TP/(TP+FN)$$

1. The F1 score is a weighted harmonic mean of precision and recall such that the best score is 1.0 and the worst is 0.0. Generally speaking, F1 scores are lower than accuracy measures as they embed precision and recall into their computation. As a rule of thumb, the weighted average of F1 should be used to compare classifier models, not global accuracy.

$$\text{F1 Score} = 2*(\text{Recall} * \text{Precision}) / (\text{Recall} + \text{Precision})$$

CHAPTER 10

ADVANTAGES AND DISADVANTAGES

ADVANTAGES

A mailbox-level anti-phishing solution offers an additional layer of protection by analyzing account information and understanding users' communication habits. This delivers an enhanced level of phishing protection to detect attacks faster, alert users and remediate threats as quickly as possible.

1. Measure the degrees of corporate and employee vulnerability.
2. Eliminate the cyber threat risk level.
3. Increase user alertness to phishing risks.
4. Instill a cyber security culture and create cyber security heroes.

DISADVANTAGES

Phishing has a list of negative effects on a business, including loss of money, loss of intellectual property, damage to reputation, and disruption of operational activities. These effects work together to cause loss of company value, sometimes with irreparable repercussions.Phishing is a problem on two fronts. First, a hacker may gain valuable access to a single account through a successful phishing attempt. Second, if an employee is using the same password for multiple company accounts, then the hacker has now gained access to a great deal of confidential company data.

CHAPTER 11

CONCLUSION

The importance to safeguard online users from becoming victims of online fraud, divulging confidential information to an attacker among other effective uses of phishing as an attacker's tool. Unfortunately, many of the existing phishing-detection tools, especially those that depend on an existing blacklist, suffer limitations such as low detection accuracy and high false alarm that is often caused by either a delay in blacklist update as a result of humanverification processinvolved in classification or perhaps, it can be attributed to human error in classification which may lead to improper classification of the classes. The inconsistent nature of attacks behaviors and continuously changing URL phish patterns require timely updating of the reference model. Therefore, it requires an effective technique to regulate retraining as to enable machine learning algorithm to actively adapt to the changes in phish patterns.Our phishing detection method focused on the learning process. We extracted 14 different features, which make phishing websites different from legitimate websites. The outcome of the experiment reached over 96% of Accuracy when websites with Logistic Regression are detected.

CHAPTER 12

FUTURE SCOPE

Phishing attacks are targeting these users depending on the trikes of social engineering. Despite there are several ways to carry out these attacks, unfortunately the current phishing detection techniques cover some attack vectors like email and fake websites.Phishing is when attackers send malicious emails designed to trick people into falling for a scam. Typically, the intent is to get users to reveal financial information, system credentials or other sensitive data.Phishing has a list of negative effects on a business, including loss of money, loss of intellectual property, damage to reputation, and disruption of operational activities. These effects work together to cause loss of company value, sometimes with irreparable repercussions.

CHAPTER 13

APPENDIX

13.1 CODE

HTML:

```
<!DOCTYPE html>
<html>
<head>
    <center><h1> IBM PROJECT BASED LEARNING</h1></center>
    <center><h2> TEAM ID : PNT2022TMID15588 </h2></center>
    <title> URL PHISHING DETECTION </title>
    <style>
        body{
            background-color:lightgrey;
        }
    </style>
</head>

<body>
    <center><h2> WEB PHISHING DETECTION </h2></center>
    <center>
    <form action="/" method="post">
        <label for="url" class="form__label"><b>Enter the URL:</b></label><br>
        <input type="text" class="form__input" name ='url' id="url" placeholder="Enter
URL" required="" />

        <button class="button" role="button" ><b>Check URL</b></button>
    </form>
        <h5><a href= {{ url }} target="_blank">{{ url }}</a></h5>
        <h3 id="prediction"></h3>
        <button class="button1" id="button1" role="button"
onclick="window.open('{{url}}')" target="_blank"><b>Continue</b></button>
        <!--<button class="button2" id="button2" role="button"
```

```
onclick="window.open('{{url}}')" target="_blank" >Still want to Continue</button>-->

    </center>

    <!-- JavaScript -->
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
      integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
      crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"
      integrity="sha384-
Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
      crossorigin="anonymous"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"
      integrity="sha384-
OgVRvuATP1z7JjHLkuOU7Xw704+h835Lr+6QL9UvYjZE3Ipu6Tp75j7Bh/kR0JKI"
      crossorigin="anonymous"></script>

    <script>
      let x = '{{xx}}';
      let num = x*100;
      if (0<=x && x<0.50){
        num = 100-num;
      }
      let txtx = num.toString();
      if(x<=1 && x>=0.50){
        var label = "Website is "+txtx +"% safe to use...";
        document.getElementById("prediction").innerHTML = label;
        document.getElementById("button2").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("button1").style.display="block";
      }
    </script>
  </body>
```

```
</html>
```

PYTHON:

```
importipaddress
import re
importurllib.request
from bs4 import BeautifulSoup
import socket
import requests
fromgooglesearch import search
importwhois
fromdatetime import date, datetime
import time
fromdateutil.parser import parse as date_parse
 fromurllib.parse import urlparse

 classFeatureExtraction:
 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
defUsingIp(self):
try:
ipaddress.ip_address(self.url)
return -1
except:
return 1

   # 2.longUrl
deflongUrl(self):
iflen(self.url) < 54:
return 1
iflen(self.url) >= 54 and len(self.url) <= 75:
return 0
return -1

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

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

'short\.to|BudURL\.com|ping\.fm|post\.ly|Just\.as|bkite\.com|snipr\.com|fic\.kr|loopt\.us|'

'doiop\.com|short\.ie|kl\.am|wp\.me|rubyurl\.com|om\.ly|to\.ly|bit\.do|t\.co|lnkd\.in|'
                'db\.tt|qr\.ae|adf\.ly|goo\.gl|bitly\.com|cur\.lv|tinyurl\.com|ow\.ly|bit\.ly|ity\.im|'

'q\.gs|is\.gd|po\.st|bc\.vc|twitthis\.com|u\.to|j\.mp|buzurl\.com|cutt\.us|u\.bb|yourls\.org|'

'x\.co|prettylinkpro\.com|scrnch\.me|filoops\.info|vzturl\.com|qr\.net|1url\.com|tweez\.me|v\.g
d|tr\.im|link\.zip\.net', self.url)
if match:
return -1
```

```
return 1

    # 4.Symbol@
def symbol(self):
ifre.findall("@",self.url):
return -1
return 1

    # 5.Redirecting//
def redirecting(self):
ifself.url.rfind('//')>6:
return -1
return 1

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

    # 7.SubDomains
defSubDomains(self):
dot_count = len(re.findall("\.", self.url))
ifdot_count == 1:
return 1
elifdot_count == 2:
return 0
return -1

    # 8.HTTPS
defHppts(self):
try:
https = self.urlparse.scheme
```

```
if 'https' in https:
return 1
return -1
except:
return 1

   # 9.DomainRegLen
defDomainRegLen(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'):
forhead.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
defNonStdPort(self):
try:
port = self.domain.split(":")
iflen(port)>1:
return -1
return 1
except:
return -1

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

# 13.RequestURL
defRequestURL(self):
try:
forimg 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

foriframe 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
defAnchorURL(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
defLinksInScriptTags(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
defServerFormHandler(self):
try:
iflen(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
defInfoEmail(self):
try:
ifre.findall(r"[mail\(\)|mailto:?]", self.soap):
return -1
else:
return 1
except:
return -1

# 18.AbnormalURL
defAbnormalURL(self):
try:
ifself.response.text == self.whois_response:
```

```
return 1
else:
return -1
except:
return -1

# 19.WebsiteForwarding
defWebsiteForwarding(self):
try:
iflen(self.response.history) <= 1:
return 1
eliflen(self.response.history) <= 4:
return 0
else:
return -1
except:
return -1

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

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

```
# 22.UsingPopupWindow
defUsingPopupWindow(self):
try:
ifre.findall(r"alert\(", self.response.text):
return 1
else:
return -1
except:
return -1

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

# 24.AgeofDomain
defAgeofDomain(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
  defDNSRecording(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
  defWebsiteTraffic(self):
  try:
  rank =
BeautifulSoup(urllib.request.urlopen("http://data.alexa.com/data?cli=10&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])
ifglobal_rank> 0 and global_rank< 100000:
return 1
return -1
except:
return -1


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

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

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

```
'at\.ua|usa\.cc|baltazarpresentes\.com\.br|pe\.hu|esy\.es|hol\.es|sweddy\.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\.27|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|103\.232\.215\.140|69\.172\.201\.153|'

  '216\.218\.185\.162|54\.225\.104\.146|103\.243\.24\.98|199\.59\.243\.120|31\.170\.160\.61|213\.19\.128\.77|62\.113\.226\.131|208\.100\.26\.234|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)
  ifurl_match:
  return -1
  elifip_match:
  return -1
  return 1
  except:
  return 1

  defgetFeaturesList(self):
  returnself.features
```

13.2 GITHUB LINK AND PROJECT DEMO LINK

GitHub Link:
https://github.com/IBM-EPBL/IBM-Project-21417-1659780033

Project Demo Link:
https://drive.google.com/file/d/1p2BjH8F_tHDfepV8rPyj-Gx-JSR1G4sr/view?usp=drivesdk