WEB PHISHING DETECTION

IBM PROJECT REPORT

Submitted by

Team ID: PNT2022TMID22986
Team Members:

DHARWIN R V J (913119104019)
THIRUKUMARAN P (913119104115)
GAJENDRAPANDI M (913119104026)
BHARATH M (913119104011)

In partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE ENGINEERING

VELAMMAL COLLEGE OF ENGINEERING AND TECHNOLOGY
MADURAI

ANNA UNIVERSITY : CHENNAI 600 025

JUNE 2022

ABSTRACT

Phishing attack is a simplest way to obtain sensitive information from innocent users. Aim of the phishers is to acquire critical information like username, password and bank account details. Cyber security persons are now looking for trustworthy and steady detection techniques for phishing websites detection. This paper deals with machine learning technology for detection of phishing URLs by extracting and analyzing various features of legitimate and phishing URLs. Decision Tree, random forest and Support vector machine algorithms are used to detect phishing websites. Aim of the paper is to detect phishing URLs as well as narrow down to best machine learning algorithm by comparing accuracy rate, false positive and false negative rate of each algorithm

1. INTRODUCTION

1.1 Project Overview

The purpose or goal behind phishing is data, money or personal information stealing through the fake website. The best strategy for avoiding the contact with the phishing web site is to detect real time malicious URL. Phishing websites can be determined on the basis of their domains. They usually are related to URL which needs to be registered (low-level domain and upper-level domain, path, query). Recently acquired status of intra-URL relationship is used to evaluate it using distinctive properties extracted from words that compose a URL based on query data from various search engines such as Google and Yahoo. These properties are further

led to the machine-learning- based classification for the identification of phishing URLs from a real dataset. This project focuses on real time URL phishing against phishing sites by using "Phish ThE Fish", an interactive and responsive website that will be used to detect whether a website is legitimate or a phishing site.

1.2 Purpose

There are a number of users who purchase products online and make payments through e-banking. There are e-banking websites that ask users to provide sensitive data such as username, password & credit card details, etc., often for malicious reasons. This type of e-banking website is known as a phishing website. Web service is one of the key communications software services for the Internet. Web phishing is one of many security threats to web services on the Internet.

Common threats of web phishing:

- Web phishing aims to steal private information, such as usernames, passwords, and credit card details, by way of impersonating a legitimate entity.

- It will lead to information disclosure and property damage.

- Large organizations may get trapped in different kinds of scams. This project mainly focuses on applying a machine-learning algorithm to detect Phishing websites.

2. LITERATURE SURVEY

2.1 Existing problem

Protecting user against phishing using Anti- phishing: -

AntiPhish is used to avoid users from using fraudulent web sites which in turn may lead to phishing attack. Here, AntiPhish traces the sensitive information to be filled by the user and alerts the user whenever he/she is attempting to share his/her information to a untrusted web site. The much effective elucidation for this is cultivating the users to approach only for trusted websites.

However, this approach is unrealistic. Anyhow, the user may get tricked. Hence, it becomes mandatory for the associates to present such explanations to overcome the problem of phishing. Widely accepted alternatives are based on the creepy websites for the identification of “clones” and maintenance of records of phishing websites which are in hit list.

Learning to Detect Phishing Emails:

An alternative for detecting these attacks is a relevant process of reliability of machine on a trait intended for the reflection of the besieged deception of user by means of electronic communication. This approach can be used in the detection of phishing websites, or the text messages sent through emails that are used for trapping the victims.

Approximately, 800 phishing mails and 7,000 non- phishing mails are traced till date and are detected accurately over 95% of them along with the categorization on the basis of 0.09% of the

genuine emails.

Phishing detection system for e-banking using fuzzy data mining: -

Phishing websites, mainly used for e-banking services, are very complex and dynamic to be identified and classified. Due to the involvement of various ambiguities in the detection, certain crucial data mining techniques may prove an effective means in keeping the e-commerce websites safe since it deals with considering various quality factors rather than exact values.

An effective approach to overcome the “fuzziness” in the e-banking phishing website assessment is used an intelligent resilient and effective model for detecting e-banking phishing websites is put forth. The applied model is based on fuzzy logics along with data mining algorithms to consider various effective factors of the ebanking phishing website.

Collaborative Detection of Fast Flux Phishing Domains:-

Here, two approaches are defined to find correlation of evidences from multiple servers of DNS and multiple suspects of FF domain. Real life examples can be used to prove that our correlation approaches expedite the detection of the FF domain, which are based on an analytical model which can quantify various DNS queries that are required to verify a FF domain.

It also shows implementation of correlation schemes on a huge level by using a distributed model, that is more scalable as compared to a centralized one, is publish N subscribe correlation model known as LARSID.

In deduction, it is quite difficult to detect the FF domains in a accurate and timely manner, as the screen of proxies is used to shield the FF Mother ship.

A theoretical approach is used to analyze the problem of FF detection by calculating the number

of DNS queries required to get back a certain amount of unique IP addresses.

A Prior-based Transfer Learning Method for the Phishing Detection: -

A logistic regression is the root of a priority based transferrable learning method, which is presented here for our classifier of statistical machine learning. It is used for the detection of the phishing websites depending on our selected characteristics of the URLs. Due to the divergence in the allocation of the features in the distinct phishing areas, multiple models are proposed for different regions.It is almost impractical to gather enough data from a new area to restore the detection model and use the transfer learning algorithm for adjusting the existing model. An appropriate way for phishing detection is to use our URL- based method.

To cope with all the prerequisites of failure of detecting characteristics, we have to adopt the transferring method to generate a more effective model.

2.2 References

• “Protecting Users Against Phishing Attacks with AntiPhish” Engin Kirda and Christopher Kruegel Technical University of Vienna

• “Learning to Detect Phishing Emails” Ian Fette School of Computer Science Carnegie Mellon University Pittsburgh, PA, 15213, USA icf@cs.cmu.edu Norman Sadeh School of Computer Science Carnegie Mellon University Pittsburgh, PA, 15213, USA Anthony Tomasic School of Computer Science Carnegie Mellon University Pittsburgh, PA, 15213, USA

• Modeling and Preventing Phishing Attacks by Markus Jakobsson, Phishing detection system for e -banking using fuzzy data mining by Aburrous, M. ; Dept. of Comput., Univ. of Bradford, Bradford, UK ; Hossain, M.A. ; Dahal, K. ; Thabatah, F.

• M. Chandrasekaran, et al., ”Phishing email detection based on structural properties”, in New York State Cyber Security Conference (NYS) , Albany, NY ,” 2006.

• P. R. a. D. L. Ganger, ”Gone phishing: Evaluating anti-phishing tools for windows. Technical report, ,” September 2006

• M. Bazarganigilani, ”Phishing E-Mail Detection Using Ontology Concept and Nave Bayes Algorithm,” International Journal of Research and Reviews in Computer Science, vol. 2,no.2, 2011

• M. Chandrasekaran, et al., ”Phoney: Mimicking user response to detect phishing attacks,” in In: Symposium on World of Wireless, Mobile and Multimedia Networks, IEEE Computer Society, 2006, pp. 668-672I.

• Fette, et al., ”Learning to detect phishing emails,” in Proc. 16th International World Wide Web Conference (WWW 2007), ACM Press, New York, NY, USA, May 2007, pp. 649-656

• A. Bergholz, et al., ”Improved phishing detection using model-based features,” in Proc. Conference on Email and Anti-Spam (CEAS).Mountain View Conf, CA, aug 2008

• L. Ma, et al.,”Detecting phishing emails using hybrid features,”IEEE Conf, 2009, pp. 493-497

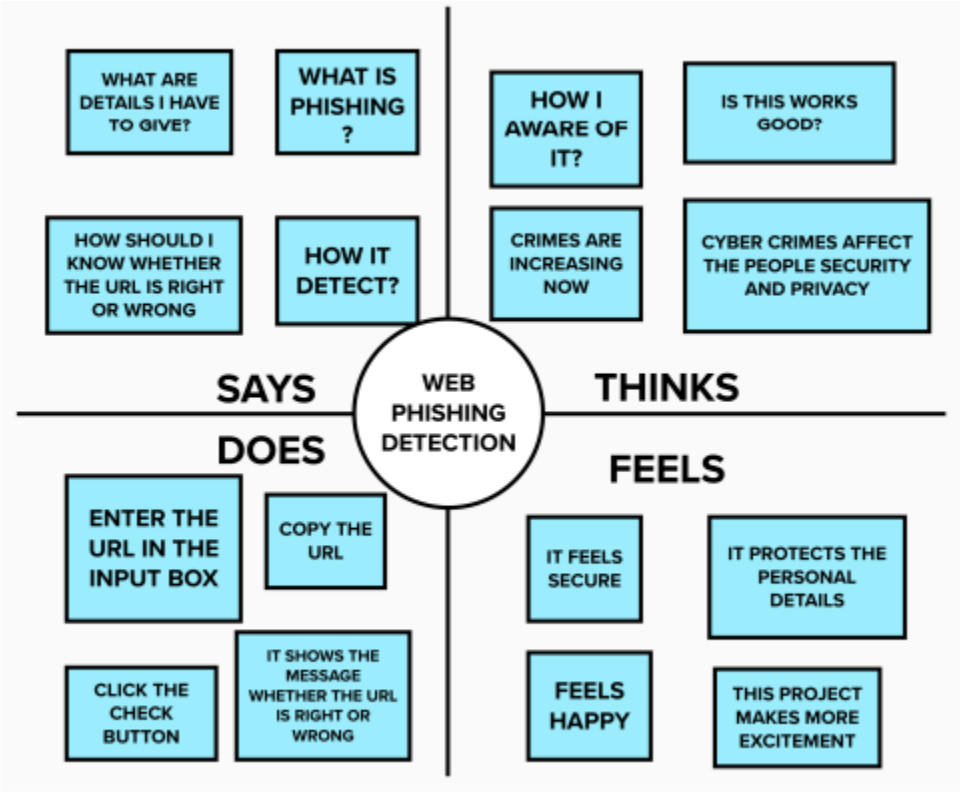
2.3 Problem Statement Definition

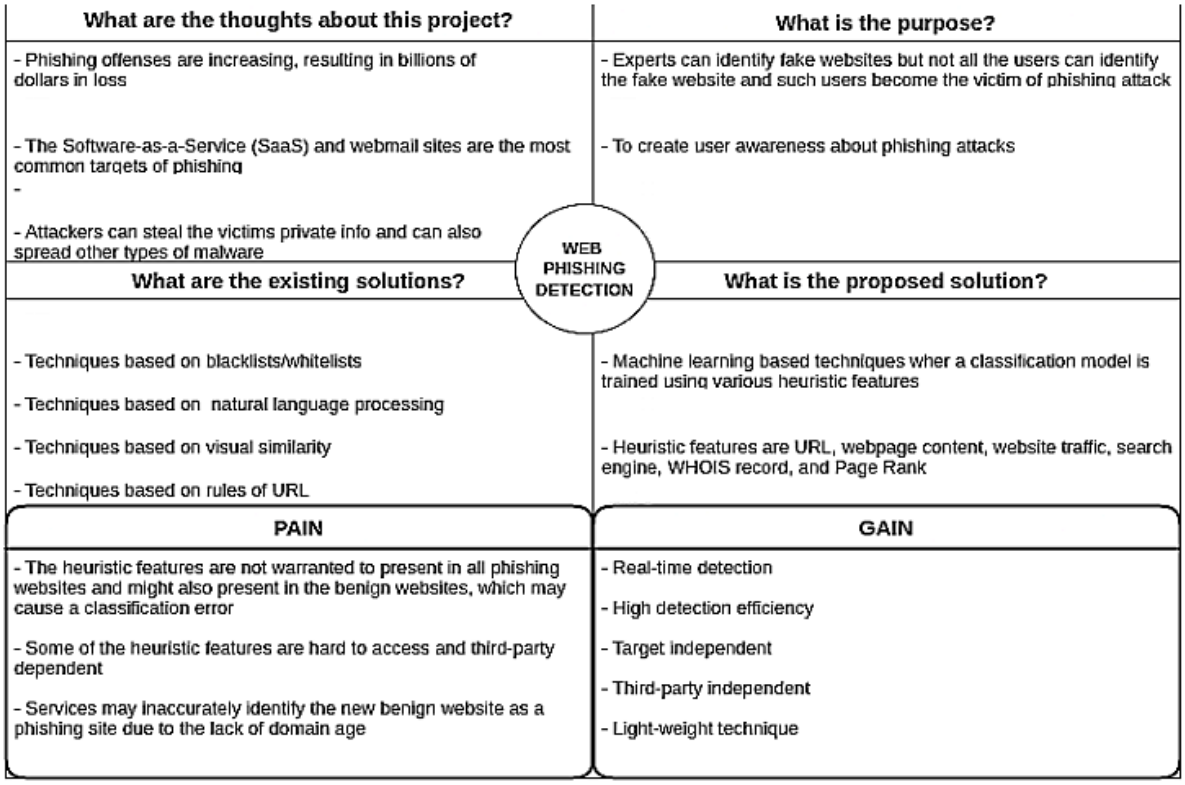
In order to detect phishing websites, we proposed an intelligent, flexible and effective system that is based on using classification algorithms. We implemented classification algorithms and techniques to extract the phishing datasets criteria to classify their legitimacy. The phishing website can be detected based on some important characteristics like URL and domain identity,

and security and encryption criteria in the final phishing detection rate. Once the user enters the URL in our site will use a data mining algorithm to detect whether the website is a phishing website or not.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas





3.2 Ideation & Brainstorming

1. Use anti-phishing protection and anti-spam software to protect yourself when malicious messages slip through to your computer.

2. Anti-spyware and firewall settings should be used to prevent phishing attacks and users should Protect your mobile phone by setting software to update automatically. These updates could give you critical protection against security threats.

3. Protect your data by backing it up. Back up your data and make sure those backups aren't connected to your home network. You can copy your computer files to an external hard drive or cloud storage. Back up the data on your phone, too.

4. The website may be used to hack and misuse others detail so to protect that Then kids nowdays are learning from online so to protect them from facing any unpleasant or bad activity. So create a extension in google which will detect the fake websites.

5. Every time you click on a link, look at the browser bar and see if matches exactly the one you would type in to go to your account.

6. All members of your executive and management team are vulnerable. If a phishing scammer acquires the email credentials of high-profile leadership, it's likely they'll target anyone they can using that very email address.

7. Almost all spam messages are malicious emails sent by unknown sources. These sources could be hackers who aim to hack into the computers of their victims.

8. Never respond to spam messages because through this, the spammer will know that the email address is active and thus, it increases the chance of your email to be constantly targeted by the spammer.

9. Do not use your personal or business email address when registering in any online contest or service such as applications, deal updates, etc. Many spammers watch these groups or emailing lists to harvest new email addresses.

10.In fact, many unsuspecting users have been dupped via text message phishing (also known as smishing) and through social media.

11.The threat of malicious messages luring users to click on a link, open a malicious webpage, download malware or provide credentials on a spoofed site proves that threat actors are getting continuously creative in their methods to hijack your assets and steal your credentials.

12.While these attacks use electronic written words to lure a user into their scam and some of the messages may be hosted in social media, a new form of messaging attacks are emerging via other cloud and SaaS (software as a service) platforms that provide in-application messaging between users.

TOP 4:

1. We would create an interactive and responsive website that will be used to detect whether a website is legitimate or phishing. This website is made using different web designing languages which include HTML, CSS, Javascript and Python.

2. It must be noted that the website is created for all users, hence it must be easy to operate with and user-friendly.

3. The website will show information regarding the services provided by us. It also contains information regarding ill- practices occurring in todays technological world.

4. The website will be created with an opinion such that people are not only able to distinguish between legitimate and fraudulent website, but also become aware of the mal-practices occrring in current world. They can stay away from the people trying to exploit ones personal information, like email address, password, debit card numbers, credit card details, CVV, bank account numbers.

3.3 Proposed Solution

Problem Statement (Problem to be solved)

There are a number of users who purchase products online and make payments through

ebanking. 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 ebanking website is known as a phishing website. Web service is one of the key communications software services for the Internet.

Idea / Solution description

Anti-spyware and firewall settings should be used to prevent phishing attacks and users should Protect your mobile phone by setting software to update automatically. The website will be created with an opinion such that people are not only able to distinguish between legitimate and fraudulent websites, but also become aware of the mal-practices occurring in the current world.

Novelty / Uniqueness

The website designed will be user friendly in means for any age. Easy to detect the fraudulent website and protect the sensitive credential information.

Social Impact / Customer Satisfaction

Feel protected by using the website as the business-related credentials will be safe. Parents can be relaxed when kids explore educational website as the fraudulent website will be detected by our website.

Business Model (Revenue Model)

This can be a efficient way to help banking sector as it secures the legitimate website from other malware that are set by hacker.

Scalability of the Solution

We would create an interactive and responsive website that will be used to detect whether a website is legitimate or phishing. This website is made using different web designing languages which include HTML, CSS, JavaScript and Python. This website is more useful to the user and it is user friendly also.

3.4 Problem Solution fit

1. CUSTOMER SEGMENT(S)

Protect yourself and your family against malicious websites with the platform for free. With the platform, protecting your staff, data, brand, and your customer from malicious websites has never been easier. Proactively protect multiple customers against malicious websites at once with all-in-one platform. The platform can be used for government embeds to provide 100% security and privacy.

2. JOBS-TO-BE-DONE / PROBLEMS

The objective of phishing website URLs is to purloin the personal information like user name, passwords and online banking transactions. Phishers use the websites which are visually and semantically similar to those real websites. As technology continues to grow, phishing techniques started to progress rapidly and this needs to be prevented by using anti-phishing mechanisms to detect phishing.

3. TRIGGERS

➤ Your users lack security awareness.

➤ Criminals are (unsurprisingly) following the money.

➤ You're not performing sufficient due diligence.

➤ Low-cost phishing and ransomware tools are easy to get hold of.

➤ Malware is becoming more sophisticated.

4. EMOTIONS: BEFORE / AFTER

➤ Greed - Clicking on fake successful messages.

➤ Urgency - Hackers use fake security alerts with exclamation marks.

➤ Helpfulness - Hackers and cybercriminals use major tragedies to appeal for help but they are only helping themselves.

➤ Fear- Emails that spread fear and phishing links go hand in hand.

5. AVAILABLE SOLUTIONS

Legitimate websites prevent web scraping by several techniques in respect to obfuscation using CSS sprites to display important data, replacing text with images.

Spam filtering techniques are used to identify unsolicited emails before the userreads or clicks the link.

When users visit a phishing web page that looks like a legitimate website, manypeople do not remember the legitimate website's domain name, particularly for somestart-ups or unknown companies, so users cannot recognize the phishing website based onthe URL. Some web

browsers integrate a security component to detect phishing or malwaresites, such as Chrome, which will display warning messages when one visits an unsafe webpage.

When the website detects that the IP address and device information of the user who is logging in does not match the commonly used information, it is necessary to verify the authenticity of the user.

6. CUSTOMER CONSTRAINTS

The limitations of the web phishing detection approaches are explored by means of detection time, detection rate, and storage complexity to verify the level of robustness against the phishing attack.

Thus most of the recent web phishing detection approaches lag in feature selection mechanism as they use handcrafted features to detect the attack.

7. BEHAVIOUR

➤ Customers should take a “trust no one” approach when opening email. ➤ Check and verify the “From” address of the email.

➤ By carefully reading the email copy, users can typically spot something that seems “off” including: An email with an “urgent” request or An email offering the user something that’s “too good to be true”.

➤ Check grammar and spelling. Poor grammar and misspelled words in an email can be red flags.

➤ Be wary of generic salutations in an email. Legitimate companies, especially those with which

you have accounts or have done business typically will address you by name versus by a generic greeting.

➤ Encourage your clients to look for any unusual or odd requests in their emails. Most fraudulent emails contain a request to respond to the email or click a link in it.

➤ Avoid clicking links or attachments in emails from unfamiliar sources.

8. CHANNELS OF BEHAVIOUR

8.1 ONLINE What kind of actions do customers take online?

Nothing teaches like experience. When employees click on a link or an attachment in a simulated phishing email, it's important to communicate to them that they have potentially put both themselves and the organization at risk.

8.2 OFFLINE What kind of actions do customers take offline?

Phishing awareness training starts with educating your employees on why phishing is harmful, and empowering them to detect and report phishing attempts.

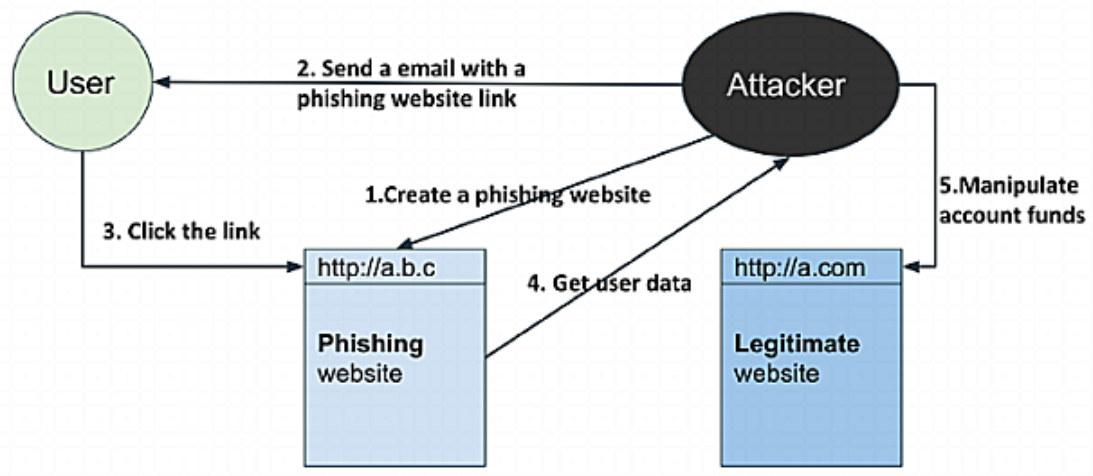
Simulated phishing campaigns reinforce employee training, and to understand risk and improve workforce resiliency as these can take many forms, such as mass phishing, spear phishing, and whaling.

9. PROBLEM ROOT CAUSE

A phishing attack is a type of cybersecurity threat that targets users directly through email, text or direct messages. During one of these scams, a cybercriminal will pose as a trusted contact to steal data from an unsuspecting user such as login information, account numbers and credit

card information.

While there are several types of phishing, the main purpose behind all of them is it to steal sensitive information or transfer malware.



10. YOUR SOLUTION

We would create an interactive and responsive website that will be used to detect whether a website is legitimate or phishing. This website is made using different web designing languages which include HTML, CSS, JavaScript and Python. This website is more useful to the user and it is user friendly also.

Problem-Solution fit canvas 2.0

Purpose / Vision

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) — CS

Who is your customer?

- Protect yourself and your family against malicious websites with the platform for free.
- With the platform, protecting your staff, data, brand, and your customer from malicious websites has never been easier.
- Proactively protect multiple customers against malicious websites at once with all-in-one platform.
- The platform can be used for government embeds to provide 100% security and privacy.

6. CUSTOMER CONSTRAINTS — CC

What constraints prevent your customers from taking action or limit their choices of solutions?

- The limitations of the web phishing detection approaches are explored by means of detection time, detection rate, and storage complexity to verify the level of robustness against the phishing attack.
- Thus most of the recent web phishing detection approaches lag in feature selection mechanism as they use handcrafted features to detect the attack.

5. AVAILABLE SOLUTIONS — AS

Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have?

- Legitimate websites prevent web scraping by several techniques in respect to obfuscation using CSS sprites to display important data, replacing text with images.
- Spam filtering techniques are used to identify unsolicited emails before the user reads or clicks the link.
- When users visit a phishing web page that looks like a legitimate website, many people do not remember the legitimate website's domain name, particularly for some start-ups or unknown companies, so users cannot recognise the phishing website based on the URL. Some web browsers integrate a security component to detect phishing or malware sites, such as Chrome, which will display warning messages when one visits an unsafe web page.
- When the website detects that the IP address and device information of the user who is logging in does not match the commonly used information, it is necessary to verify the authenticity of the user

Explore AS, differentiate

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

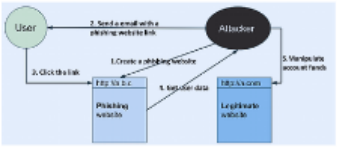
2. JOBS-TO-BE-DONE / PROBLEMS — J&P

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

- The objective of phishing website URLs is to purloin the personal information like user name, passwords and online banking transactions.
- Phishers use the websites which are visually and semantically similar to those real websites.
- As technology continues to grow, phishing techniques started to progress rapidly and this needs to be prevented by using anti-phishing mechanisms to detect phishing.

9. PROBLEM ROOT CAUSE — RC

What is the real reason that this problem exists?
What is the back story behind the need to do this job?



- A phishing attack is a type of cybersecurity threat that targets users directly through email, text or direct messages. During one of these scams, a cybercriminal will pose as a trusted contact to steal data from an unsuspecting user such as login information, account numbers and credit card information.
- While there are several types of phishing, the main purpose behind all of them is it to steal sensitive information or transfer malware.

7. BEHAVIOUR — BE

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

- Customers should take a "trust no one" approach when opening email.
- Check and verify the "From" address of the email.
- By carefully reading the email copy, users can typically spot something that seems "off" including:
- An email with an "urgent" request or An email offering the user something that's "too good to be true".
- Check grammar and spelling. Poor grammar and misspelled words in an email can be red flags.
- Be wary of generic salutations in an email. Legitimate companies, especially those with which you have accounts or have done business typically will address you by name versus by a generic greeting.
- Encourage your clients to look for any unusual or odd requests in their emails. Most fraudulent emails contain a request to respond to the email or click a link in it.
- Avoid clicking links or attachments in emails from unfamiliar sources.

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

Identify strong TR & EM

3. TRIGGERS — TR

What triggers customers to act?

- Your users lack security awareness .
- Criminals are (unsurprisingly) following the money .
- You're not performing sufficient due diligence .
- Low-cost phishing and ransomware tools are easy to get hold of .
- Malware is becoming more sophisticated .

4. EMOTIONS: BEFORE / AFTER — EM

How do customers feel when they face a problem or a job and afterwards?

- Greed- Clicking on fake successful messages.
- Urgency-Hackers use fake security alerts with exclamation marks.
- Helpfulness-Hackers and cyber criminals use major tragedies to appeal for help but they are only helping themselves.
- Fear- Emails that spread fear and phishing links go hand in hand.

10. YOUR SOLUTION — SL

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations,
solves a problem and matches customer behaviour.

- We would create an interactive and responsive website that will be used to detect whether a website is legitimate or phishing.
- This website is made using different web designing languages which include HTML, CSS, JavaScript and Python.
- This website is more useful to the user and it is user friendly also.

8. CHANNELS of BEHAVIOUR — CH

8.1 ONLINE
What kind of actions do customers take online? Extract online channels from #2

- Nothing teaches like experience. When employees click on a link or an attachment in a simulated phishing email, it's important to communicate to them that they have potentially put both themselves and the organisation at risk.

8.2 OFFLINE
What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

- Phishing awareness training starts with educating your employees on why phishing is harmful, and empowering them to detect and report phishing attempts.
- Simulated phishing campaigns reinforce employee training, and to understand risk and improve workforce resiliency as these can take many forms, such as mass phishing, spear phishing, and whaling.

Extract online & offline CH of BE

4. REQUIREMENT ANALYSIS

4.1 Functional requirements

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Registered User - Login	Login through password (Form) Login through Gmail Login through LinkedIN
FR-4	Verify the link provided by the user	User inputs the link to be verified
FR-5	Display the result	If the site link is a phishing site, user must be aware and read the precautions displayed If the site link is legit, exit the application
FR-6	Share Queries	If any doubts, send query Read FAQs

4.2 Non-Functional requirements

Non-functional Requirements:

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

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Engage the user about the process to ensure that the functionality can meet design and usability requirements.
NFR-2	**Security**	It includes intrusion prevention and detection, authentication, authorization, and confidentiality of the user information.
NFR-3	**Reliability**	It focuses on preventing failures during the lifetime of the product or system, from commissioning to decommissioning.
NFR-4	**Performance**	It is the ability of the application to always run acceptably. In time-critical scenarios, even the smallest delay in processing data can be unacceptable.
NFR-5	**Availability**	Ensuring that the application can meet its availability targets to be resilient (fault tolerance).
NFR-6	**Scalability**	It is the ability for the application to scale to meet increasing demands; for example, at peak times or as the system becomes more widely adopted.

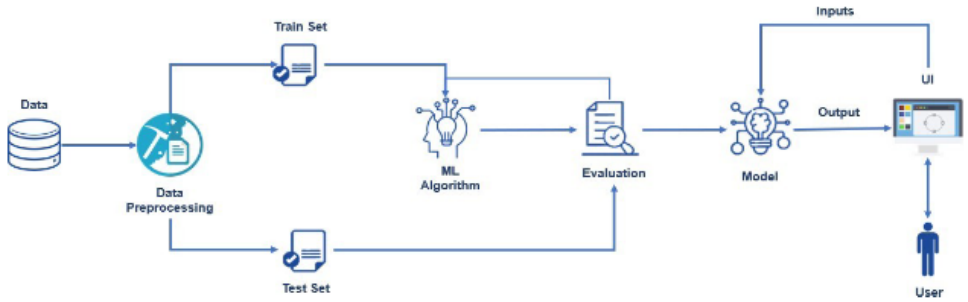
5. PROJECT DESIGN

5.1 Data Flow Diagrams

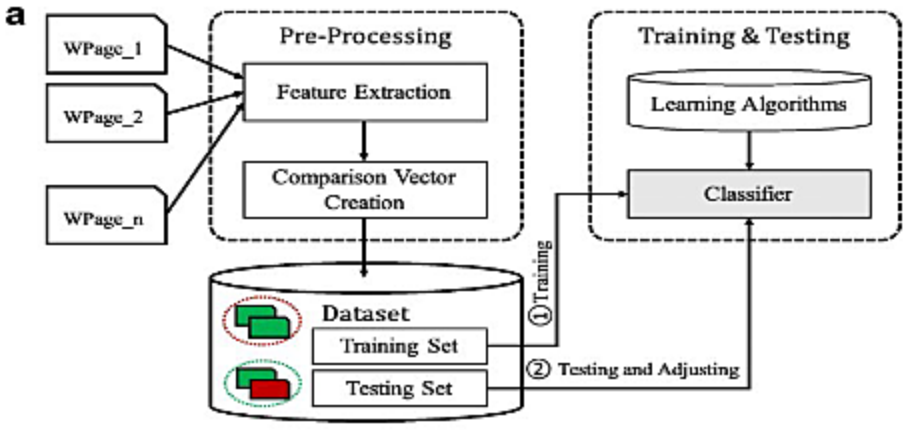
Data Flow Diagrams:

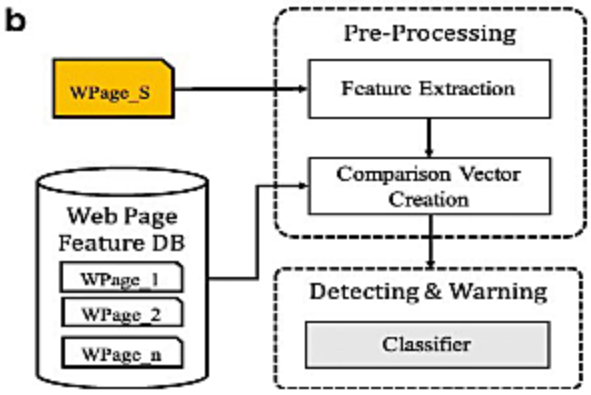
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

Architecture Diagram:



DFD Diagram:



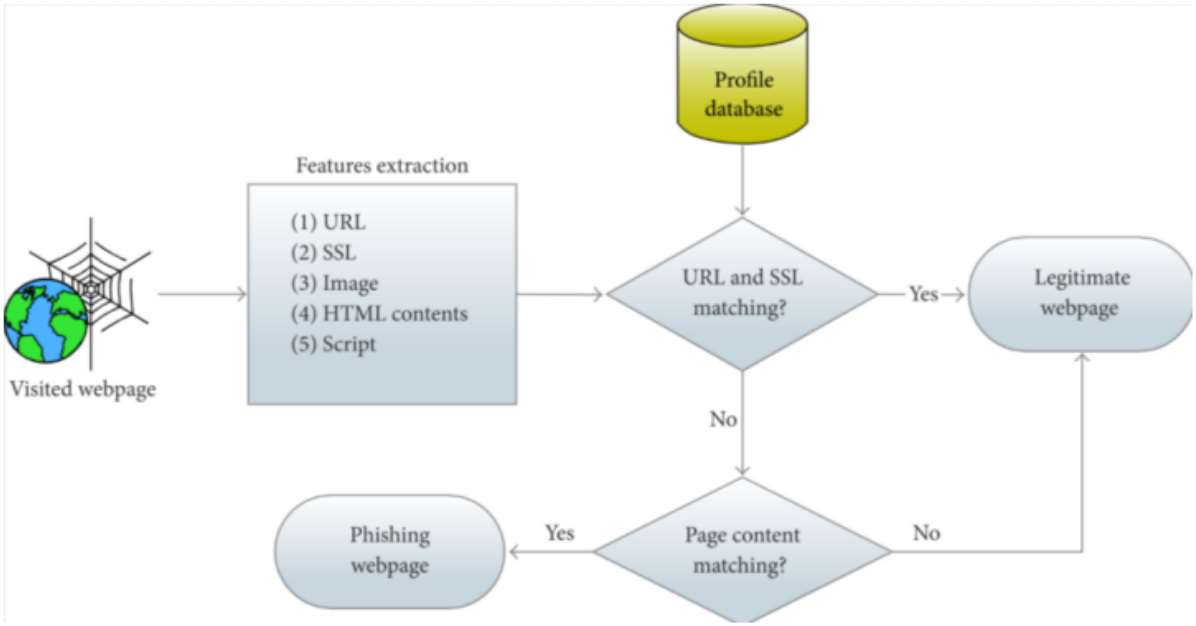


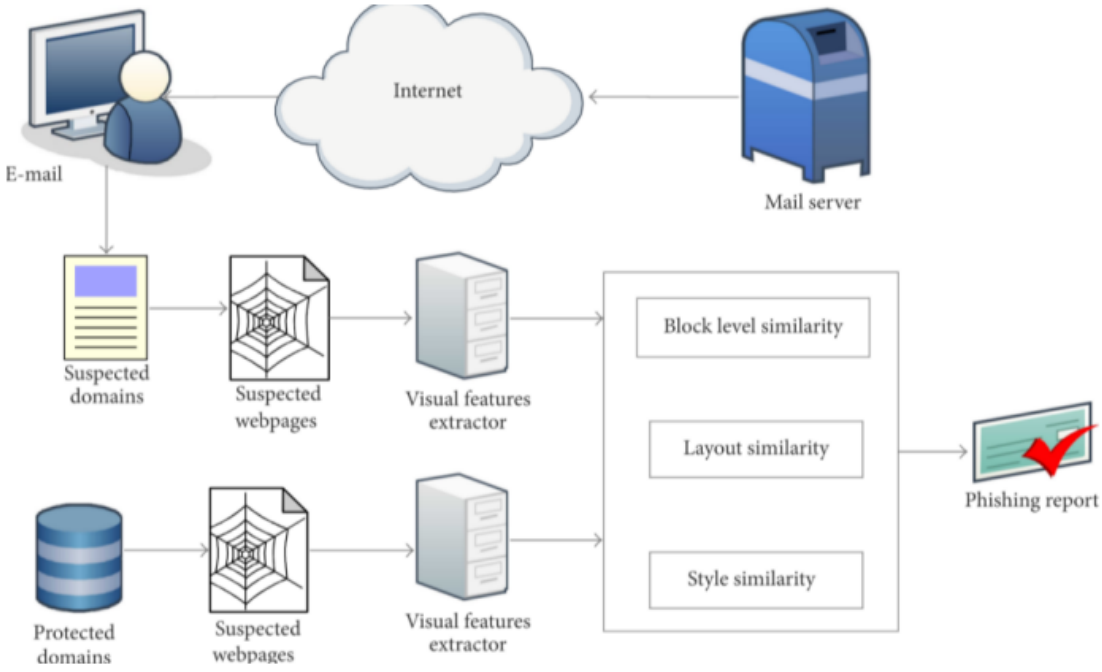
User Stories

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

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through LinkedIn	I can register & access the dashboard with LinkedIn Login	Low	Sprint-3
		USN-4	As a user, I can register for the application through Gmail	I can register & access the dashboard with Gmail Login	Medium	Sprint-2
	Login	USN-5	As a user, I can log into the application by entering email & password	I can access my account / dashboard	High	Sprint-1
	Dashboard	USN-6	As a user, I paste the Link that needs to be Verified as a Phishing site or not	I can paste the Link into the Textbox	High	Sprint-2
		USN-7	As a user,I can see the Result	I can view that it is a Safe Site	High	Sprint-2
Customer Care Executive	Help	USN-8	As a user,I can Share my Queries in the Help Textbox	I can send my Doubts through it	Medium	Sprint-3
Administrator	Contact	USN-9	As a Administrator, I can Answer the User Queries	I sent the Solution through User provided Email	Low	Sprint-4
		USN-10	As a Administrator, I can Improve the Accuracy	I can update the Website	High	Sprint-4

5.2 Solution & Technical Architecture




Internet
E-mail
Mail server
Suspected domains
Suspected webpages
Visual features extractor
Block level similarity
Layout similarity
Style similarity
Phishing report
Protected domains
Suspected webpages
Visual features extractor

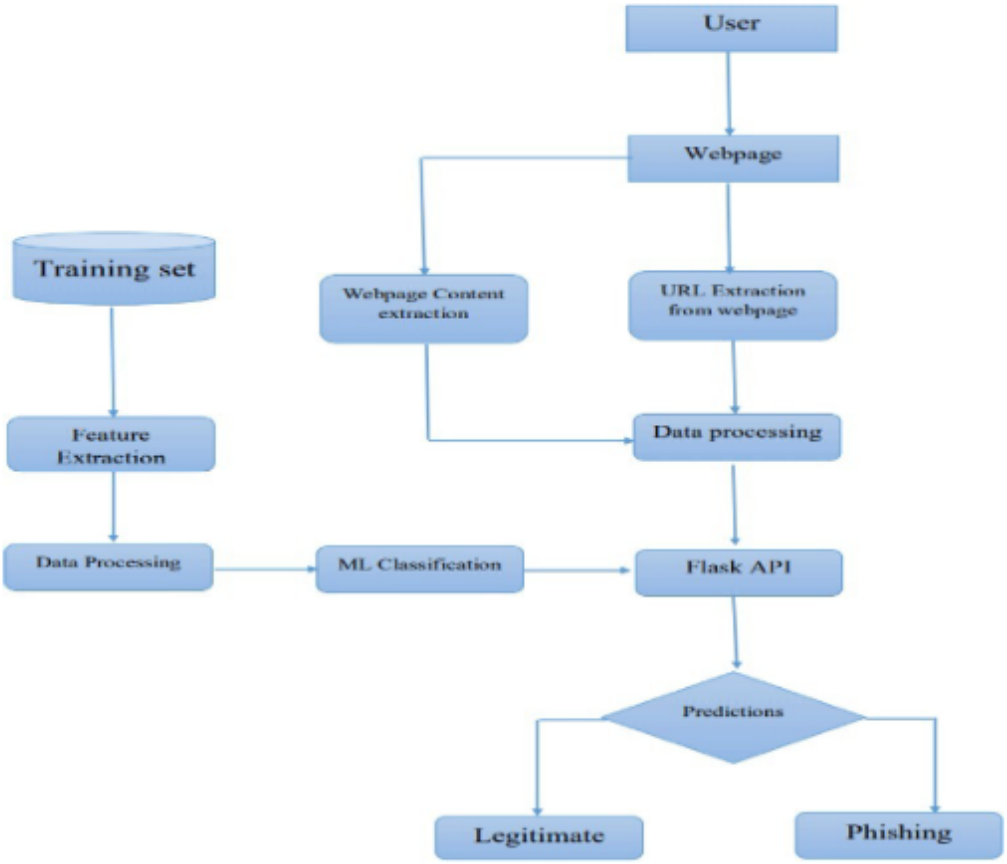

User
Webpage
Training set
Webpage Content extraction
URL Extraction from webpage
Feature Extraction
Data processing
Data Processing
ML Classification
Flask API
Predictions
Legitimate
Phishing

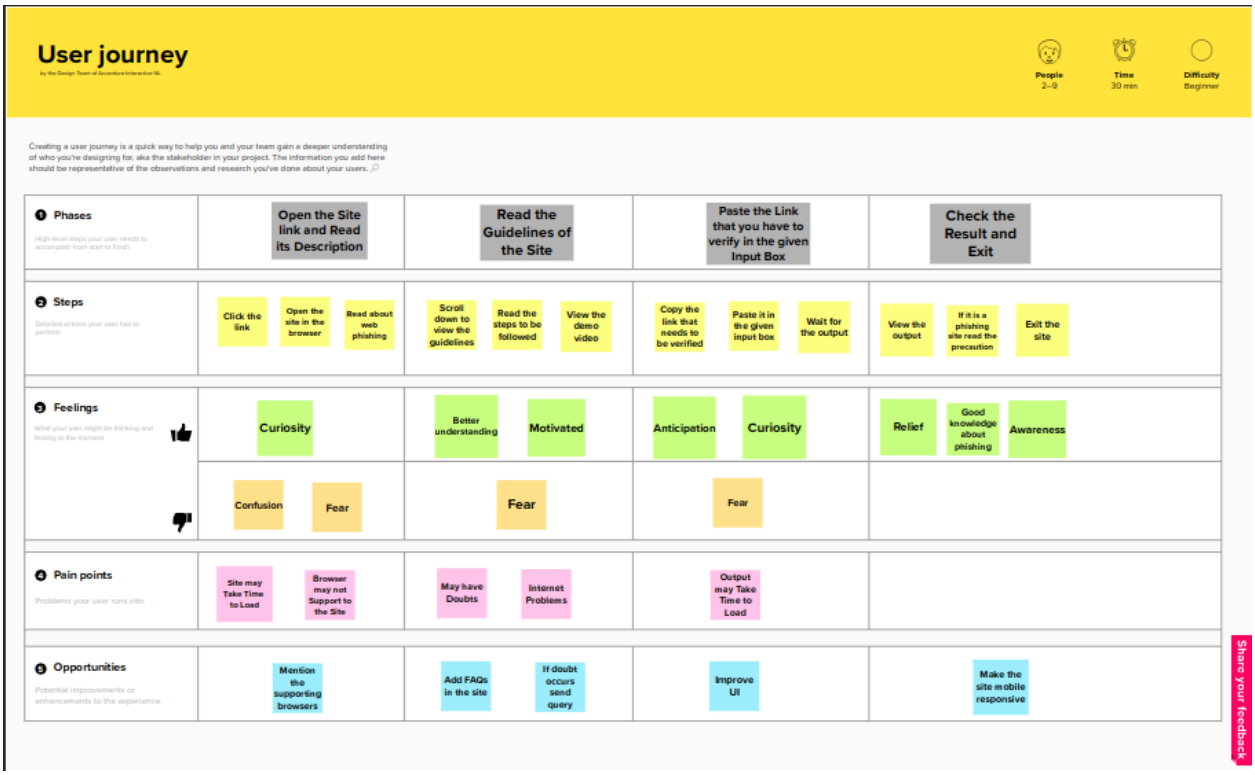
Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Dynamic Web UI	HTML, CSS, JavaScript, Bootstrap
2.	Application Logic-1	User Registration/Login	IBM API Connect Service, Gmail API, LinkedIn API
3.	Application Logic-2	Web app that predicts if the link is a phishing site or not	Flask API, Python
4.	Database	Store user input links in the database	MongoDB
5.	Cloud Database	Database Service for storing user profile	IBM DB2, IBM Cloudant etc.
6.	File Storage	Store the datasets used for prediction	Local Filesystem
7.	External API-1	User Registration/Login using email and password	IBM API Connect
8.	External API-2	User Registration/Login using external apps	Gmail API, LinkedIn API
9.	Machine Learning Model	Machine Learning Model for web phishing detection	Logistic Regression Model
10.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud	Local, Render, IBM Cloud

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	High-level open-source frameworks	Docker, Flask, Bootstrap
2.	Security Implementations	It is the security discipline that makes it possible for the right entities (people or things) to use the right resources (applications or data) when they need to, without interference, using the devices they want to use.	IAM Controls of IBM
3.	Scalable Architecture	Compose is a tool for defining and running multi-container Docker applications. With a single command, can create and start all the services from the configuration.	Docker, Docker Compose
4.	Availability	It can balance the load traffic among the servers to help improve uptime. Can scale applications by adding or removing servers, with minimal disruption to traffic flows.	IBM Cloud load balancers
5.	Performance	It provides performance feedback such as page size and how long it takes to load a page, and can show the impact new features have on the performance of the site.	IBM's SpeedCurve and Delivery Pipeline

5.3 User Stories



6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	5	High	Dharwin R V J
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	5	High	Dharwin R V J
Sprint-3		USN-3	As a user, I can register for the application through LinkedIn	10	Low	Dharwin R V J
Sprint-2		USN-4	As a user, I can register for the application through Gmail	5	Medium	Dharwin R V J
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	10	High	Dharwin R V J
Sprint-2	Dashboard	USN-6	As a user, I paste the Link that needs to be Verified as a Phishing site or not	5	High	Dharwin R V J
Sprint-2		USN-7	As a user,I can see the Result	10	High	Dharwin R V J
Sprint-3	Help	USN-8	As a user,I can Share my Queries in the Help Textbox	10	Medium	Dharwin R V J
Sprint-4	Contact	USN-9	As a Administrator, I can Answer the User Queries	10	Low	Dharwin R V J
Sprint-4		USN-10	As a Administrator, I can Improve the Accuracy	10	High	Dharwin R V J

6.2 Sprint Delivery Schedule

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

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	15	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	10	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 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)

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

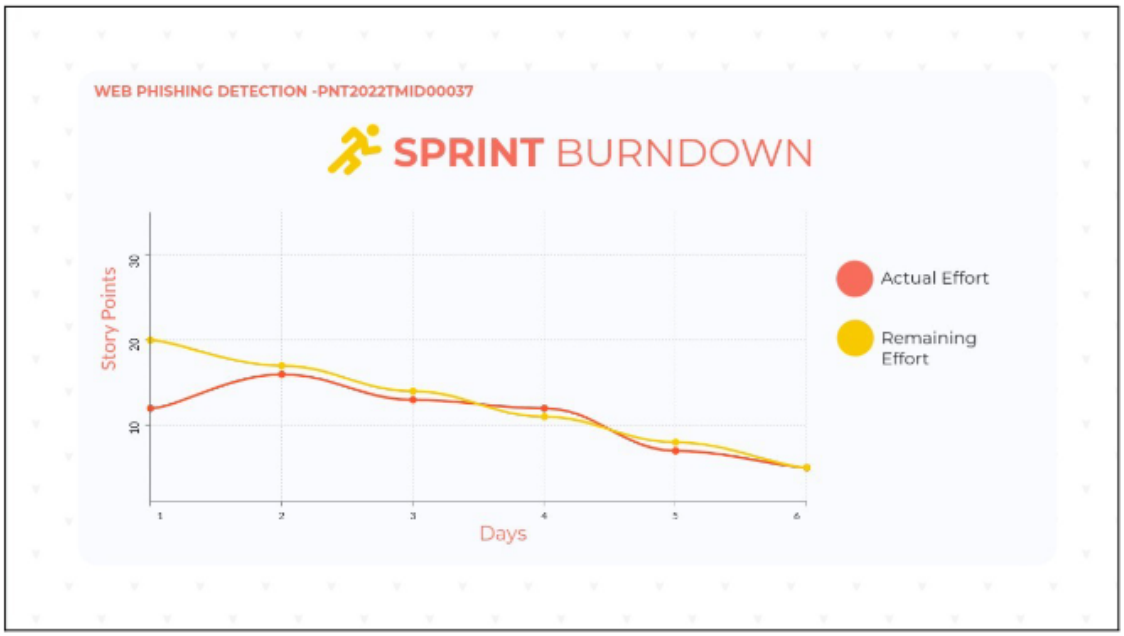
We have a 6-day sprint duration, and the velocity of the team is 20 (points per sprint). So our team's average velocity (AV) per iteration unit (story points per day)

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

Burndown Chart:

A burndown 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.

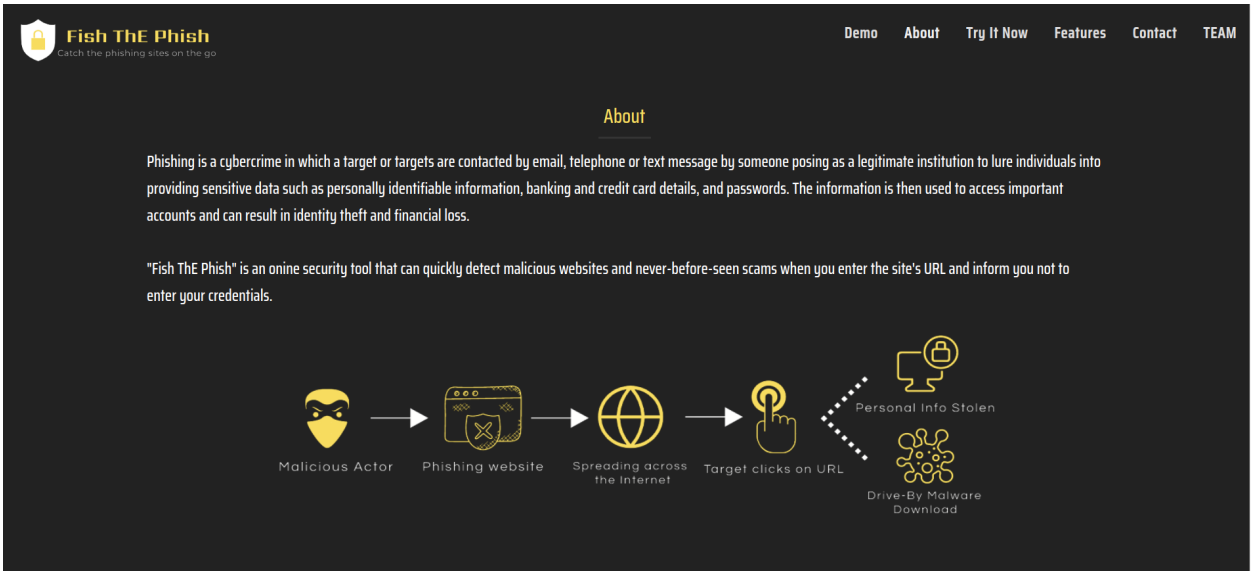
6.3 Reports from JIRA



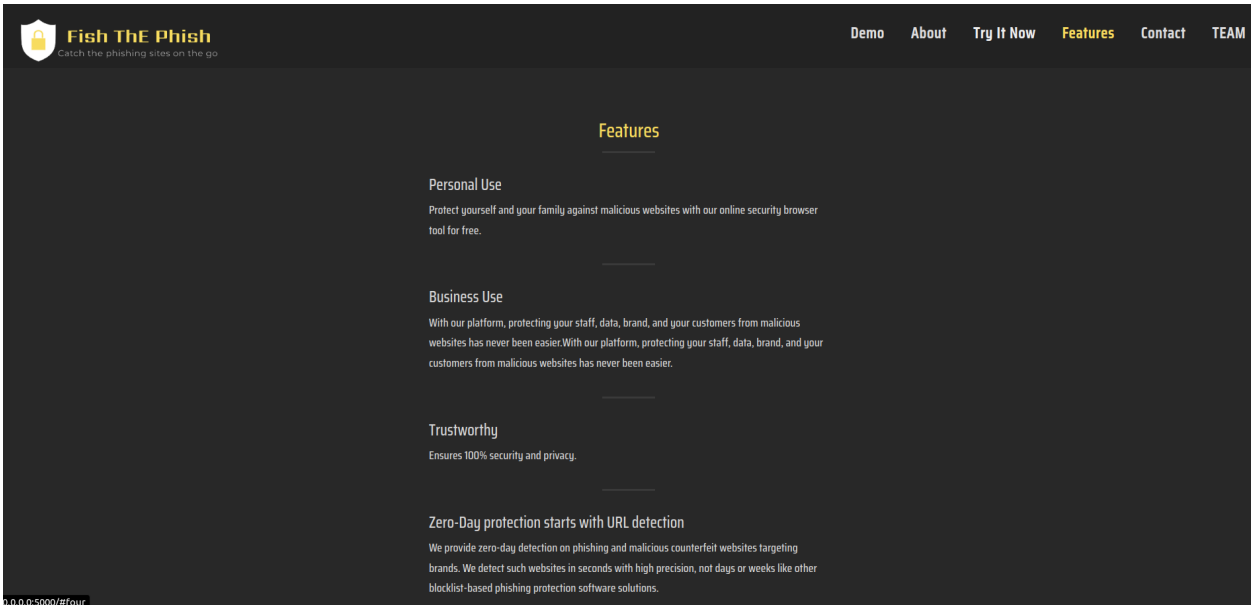
7. CODING & SOLUTIONING

7.1 Feature 1

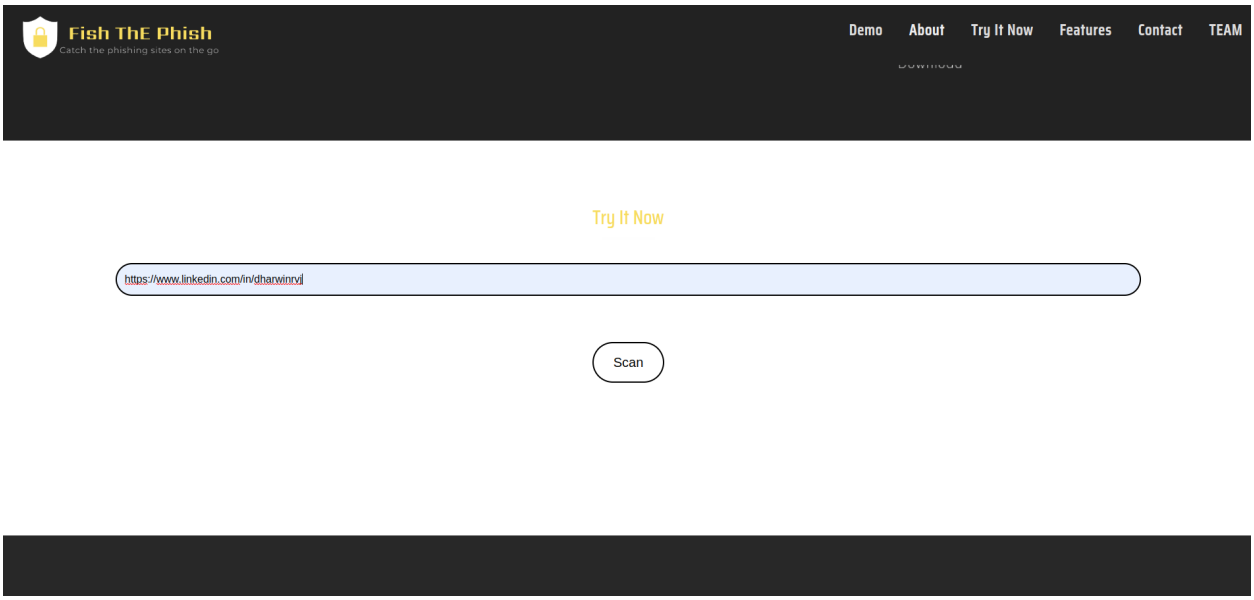
7.2 Feature 2



7.3 Feature 3



7.4 DEMO



Try It Now

Paste the URL

Scan

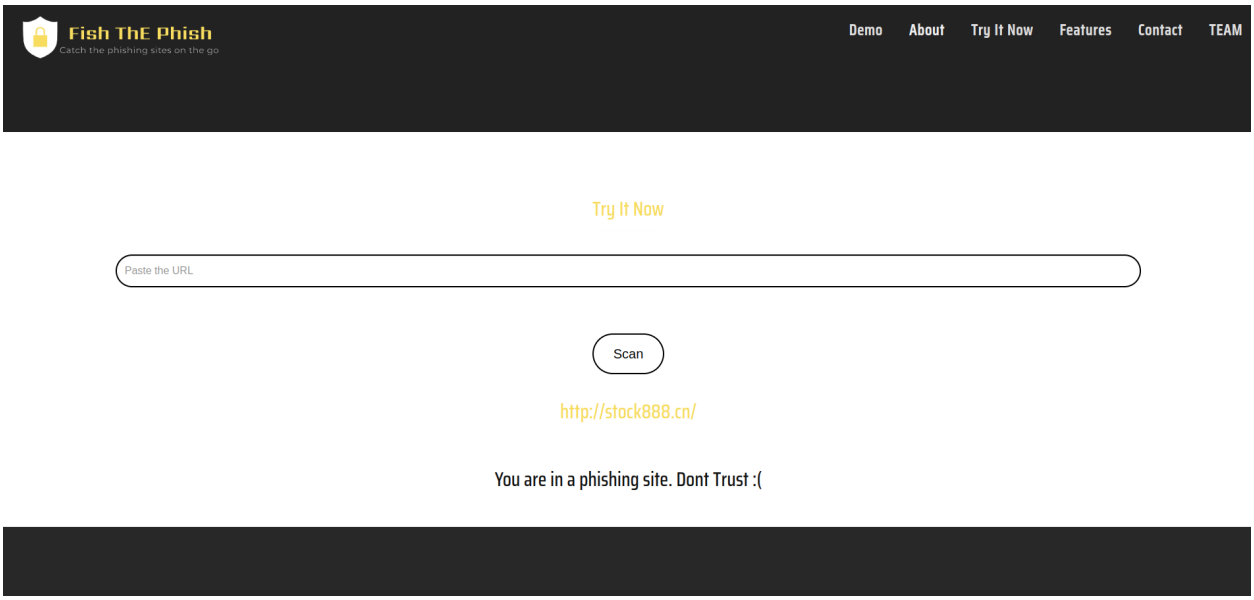
https://www.linkedin.com/in/dharwinrvj

You are safe!! This is a Legimate Website :)

Try It Now

http://stock888.cn/

Scan



8. TESTING

8.1 Test Cases

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	BUG ID
LoginPage_TC_OO1	Functional	Home Page	Verify user is able to see the Landing Page when user can type the URL in the box		1.Enter URL and click go 2.Type the URL 3.Verify whether it is processing or not.	Should Display the Webpage	Working as expected	Pass		N	
LoginPage_TC_OO2	UI	Home Page	Verify the UI elements is Responsive		1.Enter URL and click go 2. Type or copy paste the URL 3. Check whether the button is responsive or not 4. Reload and Test Simultaneously	Should Wait for Response and then gets Acknowledge	Working as expected	Pass		N	
LoginPage_TC_OO3	Functional	Home page	Verify whether the link is legitimate or not		1.Enter URL and click go 2. Type or copy paste the URL 3. Check the website is legitimate or not 4. Observe the results	User should observe whether the website is legitimate or not.	Working as expected	Pass		N	
LoginPage_TC_OO4	Functional	Home Page	Verify user is able to access the legitimate website or not		1.Enter URL and click go 2. Type or copy paste the URL 3. Check the website is legitimate or not 4. Continue if the website is legitimate or be cautious if it is not legitimate.	Application should show that Safe Webpage or Unsafe.	Working as expected	Pass		N	
LoginPage_TC_OO5	Functional	Home Page	Testing the website with multiple URLs		1.Enter URL (https://phishingshield.herokuapp.com/) and click go 2. Type or copy paste the URL to test 3. Check the website is legitimate or not 4. Continue if the website is secure or be cautious if it is not secure	User can able to identify the websites whether it is secure or not	Working as expected	Pass		N	

8.2 User Acceptance Testing

2. Defect Analysis

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	77

3. Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

9. RESULTS

9.1 Performance Metrics

```
In [25]:
score = [log_reg,ran_for,des_class,kn_class,supp_vec]
Models = pd.DataFrame({
    'Classfication Algorithms': ["Logistic Regression","Random Forest Classifier","Decision Tree Classifier", "K Neighbors
    'Accuracy': score})
Models.sort_values(by='Accuracy', ascending=False)
```

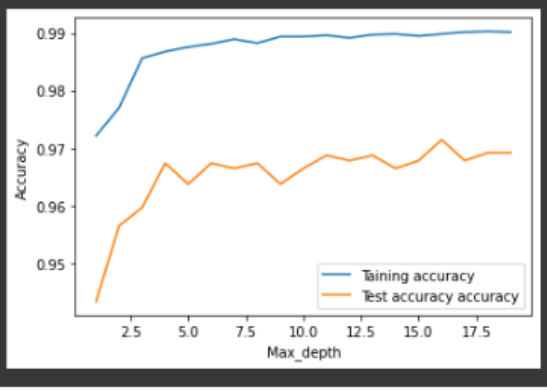
Out[25]:

	Classfication Algorithms	Accuracy
1	Random Forest Classifier	0.969697
2	Decision Tree Classifier	0.963817
3	K Neighbors Classifier	0.943464
4	Support Vector Machine	0.940751
0	Logistic Regression	0.916780

Performance:

```
[25] import matplotlib.pyplot as plt

    training_accuracy=[]
    test_accuracy=[]
    depth = range(1,20)
    for n in depth:
      rfc = RandomForestClassifier(n_estimators=n)
      rfc.fit(x_train,y_train)
      training_accuracy.append(rfc.score(x_train,y_train))
      test_accuracy.append(rfc.score(x_test,y_test))
    plt.figure(figsize=None)
    plt.plot(depth,training_accuracy,label="Taining accuracy")
    plt.plot(depth,test_accuracy,label="Test accuracy accuracy")
    plt.ylabel("Accuracy")
    plt.xlabel("Max_depth")
    plt.legend();
```



2. Tune the model

```
from sklearn.metrics import r2_score
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error

print ('R Squared =',r2_score(y_test, y_pred2))
print ('Mean Absolute Error =',mean_absolute_error(y_test, y_pred2))
print ('Mean Square Error =',mean_squared_error(y_test, y_pred2))
```

```
R Squared = 0.8761301676281433
Mean Absolute Error = 0.06151062867480778
Mean Square Error = 0.12302125734961555
```

10. ADVANTAGES & DISADVANTAGES

ADVANTAGES

STOP PHISHING AT THE E-MAIL LEVEL

The most popular means phishers adopt to trick the end-users is by sending emails across the internet and asking for web user's bank account login details. These mails are sent with the intention of making it look legal in the sight of the users and making them believe it is from a trusted sites. The users are simply asked to visit a fake websites where they will be asked to input their login credentials and thereafter reap them of their financial benefits. As this is done, there is an approach that can be taken to stop these mails from getting to the users. According to Viswanath et al. (2011) the more e-mails that one receives, the more likely he is to be deceived. The risk is comparatively higher for the ones who not only receive but also respond to

a large volume of e-mails. Organizations have responsibilities in protecting customers and employees (Users) by setting up spam filters that would categorize the emails into illegal and legal. With this kind of tool(spam filters), suspicious phishing e-mails are prevented from getting to their destinations(users).A lot of people has proposed means by which fraudulent emails can be stopped.(Garfinkel et al 2005 also suggested that internet users should adopt digitally signed mail as countermeasures for phishing e-mails. These digitally signed emails is encrypted and and uniquely identifies the sender.

SECURITY AND PASSWORD MANAGEMENT TOOLBARS

Passwords are meant to protect the accounts of users, but unfortunately most users give them away easily. Some users just because they don't want the headache of putting so many passwords in mind, decides to use a passwords for multiple account which makes it easier for phishers. Gouda et al 2007 proposed antiphishing single password protocol that allows a user to securely use a single password across multiple servers and also prevents phishing attacks. The users' computers may contain some software based protection that manages the users' passwords but they ignorantly disregard the functionality due to lack of knowledge.

VISUALLY DIFFERENTIATE THE PHISHING SITES

To help users differentiate between legal and illegal site, a dynamic security skins (DSS) a new class of human interactive proofs (HIPs) was proposed (Liu et al 2006). (Dhamija and Tygar,2005). DSS allows a remote web server to prove its identity in a way that is easy for a human user to verify and difficult for attacker. The user is able to identify its personal image and only inputs password when the image displays. If users fails to differentiate between an HTTP and a HTTPs session either due to ignorance, the proposed method is defeated.

ANTI-PHISHING TRAINING

This is the center of it all as it actively protect users from phishing threats. It is evident that phishing attacks is getting advanced even to the nearest future. Organizations needs to educate their employees on the potential risks of phishing. As technology increases, and become more universal, human remains the most vulnerable target for phishers. Training users on how best to respond to phishing attacks can reduce the success rate of the phishers. These trainings should be continuous as users tend to forget over time and the need to get updated to phishing techniques. Although educating user seem effective it cannot completely cure phishing attacks

LEGAL SOLUTION

Since Phishing has become part of the society and technology advancement, it is recommended that necessary legislation be put to place. Mcnealy 2008 examines the existing state laws in US aimed at stopping phishing attacks and the proposed federal legislation. He concluded that proper legal solutions would enable severe punishment on those caught phishing. By this, phishers are careful in their attacks. Victims of phishing attacks are also allowed to claim damage.

VISUAL BASED SIMILARITY

This operates on the principle of Visual Similarity Based Phishing Technique (VSBPT). Fishers usually like to imitate genuine websites that a lot of victim usually visit. During this process they try as much as possible for complete resemblance. The techniques they usually use are the font size, text and how the images of the genuine website appear. In as much as fishers are able to imitate the genuine website, sometimes there are no complete resemblance. With this the fake

website is usually uses the same characteristics used by the phishers to imitate the original and when the slightest difference is spotted, then the user is given a warning that the website is a phishing website.

DISADVANTAGES

ANTIMALWARE

Antimalware is now common in almost all organizations as it is used mainly in controlling phishing attack. However, most organizations have either weak antimalware or ones that are not up-to-date. Malware writers keep on altering the structure of the malware therefore antimalwares are either rewritten or updated regularly to combat the new types of malwares that are written consistently.

COMMUNICATION BETWEEN PARTIES

Most organizations have clients or customers whom they provide services to especially the financial institutions. Phishers are usually motivated by money and therefore they tend to attack financial institutions usually their clients. It is important that these institutions have consistent communication with their clients so that if one person gets attacked, they can get the information quickly and get to warn or pass the message to the other clients to prevent an extensive damage. This can also be done by the Abuse system where clients report all phishing mail to the organization and the other clients.

PHISHING INCIDENT MANAGEMENT POLICY

A phishing incident management policy should be provided by all organizations. This should be made known to users and customers. They should be trained and educated on the specific

responsibilities that will be expected from them. The policy shall be updated regularly because as technologies are changing, phishing is also taking a different trend always. It should be a comprehensive management policy that can address all the problems associated with phishing including those encountered and those not encountered. This policy shall be made ready at all times and tested regularly. It is important to test the policies regularly to detect vulnerabilities so that when the controls are not effective it shall be made known and changed.

11. CONCLUSION

This project aims to enhance detection method to detect phishing websites using machine learning technology. We achieved 96.69% detection accuracy using random forest algorithm with lowest false positive rate. In future hybrid technology will be implemented to detect phishing websites more accurately, for which random forest algorithm of machine learning technology and blacklist method will be used.

12. FUTURE SCOPE

Phishing detection is now an area of great interest among the researchers due to its significance in protecting privacy and providing security. There are many methods that perform phishing detection by classification of websites using trained machine learning models. URL based analysis increases the speed of detection. Furthermore, by applying feature selection algorithms and dimensionality reduction techniques, we can reduce the number of features and remove irrelevant data. There are many machine learning algorithms that perform classification

with good performance measures. This will serve as a guide for new researchers to understand the process and proceed to achieve better accuracy and performance.

13. APPENDIX :

Source Code: Source Code

app.py

```
import numpy as np
from flask import Flask, render_template, request, redirect, jsonify
from markupsafe import escape
import pickle
import inputScript   #inputScript file - to analyze the URL
import os
from pathlib import Path

app = Flask(__name__)

# current_directory = Path(".")
# file = os.path.join(current_directory,'phishing_website.pkl')

model = pickle.load(open('phishing_website.pkl','rb'))

# user-inputs the URL in this page
@app.route('/')
def predict():
    return render_template("final.html")

#  fetches given URL and passes to inputScript
@app.route('/predict',methods=["POST"])
def y_predict():
    url = request.form['url']
```

```
    check_predic = inputScript.main(url)
    predic = model.predict(check_predic)

    # print(check_predic)
    # print (predic)
    # result = predic[0]

    if(predic==-1):
        pred = "You are safe!! This is a Legimate Website :)"
    elif(predic==1):
        pred = "You are in a phishing site. Dont Trust :("
    else:
        pred = "You are in a suspecious site. Be Cautious ;("

    return render_template("final.html", pred_text = '{}'.format(pred),
url = url)

#  takes ip parameters from URL by inputScript and returns the predictions
@app.route('/predict_api', methods = ['POST'])
def predict_api():

    data = request.get_json(force = True)
    predic = model.y_predict([np.array(list(data.values()))])
    result = predic[0]
    return jsonify(result)

if __name__ == "__main__":
    app.run(host = '0.0.0.0', debug=True)
```

inputScript.py

```
import regex
from tldextract import extract
```

```
import ssl
import socket
from bs4 import BeautifulSoup
import urllib.request
import whois
import datetime


def url_having_ip(url):
#using regular function
#    symbol =
regex.findall(r'(http((s)?)://)((((\d)+).)*)((\w)+)(/((\w)+))?',url)
 #   if(len(symbol)!=0):
  #      having_ip = 1 #phishing
   # else:
    #    having_ip = -1 #legitimate
    #return(having_ip)
    return 0


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


def url_short(url):
    #ongoing
    return 0


def having_at_symbol(url):
    symbol=regex.findall(r'@',url)
```

```
    if(len(symbol)==0):
        return -1
    else:
        return 1

def doubleSlash(url):
    #ongoing
    return 0

def prefix_suffix(url):
    subDomain, domain, suffix = extract(url)
    if(domain.count('-')):
        return 1
    else:
        return -1

def sub_domain(url):
    subDomain, domain, suffix = extract(url)
    if(subDomain.count('.')==0):
        return -1
    elif(subDomain.count('.')==1):
        return 0
    else:
        return 1

def SSLfinal_State(url):
    try:
#check wheather contains https
        if(regex.search('^https',url)):
            usehttps = 1
        else:
            usehttps = 0
#getting the certificate issuer to later compare with trusted issuer
        #getting host name
        subDomain, domain, suffix = extract(url)
        host_name = domain + "." + suffix
```

```
        context = ssl.create_default_context()
        sct = context.wrap_socket(socket.socket(), server_hostname =
host_name)
        sct.connect((host_name, 443))
        certificate = sct.getpeercert()
        issuer = dict(x[0] for x in certificate['issuer'])
        certificate_Auth = str(issuer['commonName'])
        certificate_Auth = certificate_Auth.split()
        if(certificate_Auth[0] == "Network" or certificate_Auth ==
"Deutsche"):
            certificate_Auth = certificate_Auth[0] + " " +
certificate_Auth[1]
        else:
            certificate_Auth = certificate_Auth[0]
        trusted_Auth =
['Comodo','Symantec','GoDaddy','GlobalSign','DigiCert','StartCom','Entrust
','Verizon','Trustwave','Unizeto','Buypass','QuoVadis','Deutsche
Telekom','Network
Solutions','SwissSign','IdenTrust','Secom','TWCA','GeoTrust','Thawte','Dos
ter','VeriSign']
#getting age of certificate
        startingDate = str(certificate['notBefore'])
        endingDate = str(certificate['notAfter'])
        startingYear = int(startingDate.split()[3])
        endingYear = int(endingDate.split()[3])
        Age_of_certificate = endingYear-startingYear

#checking final conditions
        if((usehttps==1) and (certificate_Auth in trusted_Auth) and
(Age_of_certificate>=1) ):
            return -1 #legitimate
        elif((usehttps==1) and (certificate_Auth not in trusted_Auth)):
            return 0 #suspicious
        else:
            return 1 #phishing

```

```
    except Exception as e:
        return 1

def domain_registration(url):
    try:
        w = whois.whois(url)
        updated = w.updated_date
        exp = w.expiration_date
        length = (exp[0]-updated[0]).days
        if(length<=365):
            return 1
        else:
            return -1
    except:
        return 0

def favicon(url):
    #ongoing
    return 0

def port(url):
    #ongoing
    return 0

def https_token(url):
    subDomain, domain, suffix = extract(url)
    host =subDomain +'.' + domain + '.' + suffix
    if(host.count('https')): #attacker can trick by putting https in
domain part
        return 1
    else:
        return -1

def request_url(url):
    try:
```

```
        subDomain, domain, suffix = extract(url)
        websiteDomain = domain

        opener = urllib.request.urlopen(url).read()
        soup = BeautifulSoup(opener, 'lxml')
        imgs = soup.findAll('img', src=True)
        total = len(imgs)

        linked_to_same = 0
        avg =0
        for image in imgs:
            subDomain, domain, suffix = extract(image['src'])
            imageDomain = domain
            if(websiteDomain==imageDomain or imageDomain==''):
                linked_to_same = linked_to_same + 1
        vids = soup.findAll('video', src=True)
        total = total + len(vids)

        for video in vids:
            subDomain, domain, suffix = extract(video['src'])
            vidDomain = domain
            if(websiteDomain==vidDomain or vidDomain==''):
                linked_to_same = linked_to_same + 1
        linked_outside = total-linked_to_same
        if(total!=0):
            avg = linked_outside/total

        if(avg<0.22):
            return -1
        elif(0.22<=avg<=0.61):
            return 0
        else:
            return 1
    except:
        return 0
```

```
def url_of_anchor(url):
    try:
        subDomain, domain, suffix = extract(url)
        websiteDomain = domain

        opener = urllib.request.urlopen(url).read()
        soup = BeautifulSoup(opener, 'lxml')
        anchors = soup.findAll('a', href=True)
        total = len(anchors)
        linked_to_same = 0
        avg = 0
        for anchor in anchors:
            subDomain, domain, suffix = extract(anchor['href'])
            anchorDomain = domain
            if(websiteDomain==anchorDomain or anchorDomain==''):
                linked_to_same = linked_to_same + 1
        linked_outside = total-linked_to_same
        if(total!=0):
            avg = linked_outside/total

        if(avg<0.31):
            return -1
        elif(0.31<=avg<=0.67):
            return 0
        else:
            return 1
    except:
        return 0

def Links_in_tags(url):
    try:
        opener = urllib.request.urlopen(url).read()
        soup = BeautifulSoup(opener, 'lxml')

        no_of_meta =0
```

```
        no_of_link =0
        no_of_script =0
        anchors=0
        avg =0
        for meta in soup.find_all('meta'):
            no_of_meta = no_of_meta+1
        for link in soup.find_all('link'):
            no_of_link = no_of_link +1
        for script in soup.find_all('script'):
            no_of_script = no_of_script+1
        for anchor in soup.find_all('a'):
            anchors = anchors+1
        total = no_of_meta + no_of_link + no_of_script+anchors
        tags = no_of_meta + no_of_link + no_of_script
        if(total!=0):
            avg = tags/total

        if(avg<0.25):
            return -1
        elif(0.25<=avg<=0.81):
            return 0
        else:
            return 1
    except:
        return 0

def sfh(url):
    #ongoing
    return 0

def email_submit(url):
    try:
        opener = urllib.request.urlopen(url).read()
        soup = BeautifulSoup(opener, 'lxml')
        if(soup.find('mailto:')):
            return 1
```

```
        else:
            return -1
    except:
        return 0

def abnormal_url(url):
    #ongoing
    return 0

def redirect(url):
    #ongoing
    return 0

def on_mouseover(url):
    #ongoing
    return 0

def rightClick(url):
    #ongoing
    return 0

def popup(url):
    #ongoing
    return 0

def iframe(url):
    #ongoing
    return 0

def age_of_domain(url):
    try:
        w = whois.whois(url)
        start_date = w.creation_date
        current_date = datetime.datetime.now()
        age =(current_date-start_date[0]).days
        if(age>=180):
```

```
            return -1
        else:
            return 1
    except Exception as e:
        print(e)
        return 0

def dns(url):
    #ongoing
    return 0

def web_traffic(url):
    #ongoing
    return 0

def page_rank(url):
    #ongoing
    return 0

def google_index(url):
    #ongoing
    return 0

def links_pointing(url):
    #ongoing
    return 0

def statistical(url):
    #ongoing
    return 0

def main(url):

```

```
    check =
[[url_having_ip(url),url_length(url),url_short(url),having_at_symbol(url),
doubleSlash(url),prefix_suffix(url),sub_domain(url),SSLfinal_State(url),
domain_registration(url),favicon(url),port(url),https_token(url),request_u
rl(url),
url_of_anchor(url),Links_in_tags(url),sfh(url),email_submit(url),abnormal_
url(url),
redirect(url),on_mouseover(url),rightClick(url),popup(url),iframe(url),
age_of_domain(url),dns(url),web_traffic(url),page_rank(url),google_index(u
rl),
              links_pointing(url),statistical(url)]]
    # print(check)
    return check
```

web-phishing-detec.py

```
# %%
#import required libs
import pandas as pd
import numpy as np
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import confusion_matrix
# %%
#read the dataset
ds = pd.read_csv("dataset_website.csv")
```

```
ds.head()
```

```
# %%
#DATA PRE-PROCESSING
#check for null values
ds.info()
ds.isnull().any()
```

```
# %%
#split data as indep(x-all cols) and dep(y-Resut col)
#remove index col in indep ds(31 cols)
x = ds.iloc[:,1:31].values
y = ds.iloc[:,-1].values
print(x,y)
```

```
# %%
#splitting dataset into train and test ds
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test =
train_test_split(x,y,test_size=0.2,random_state=0)
```

```
# %%
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
```

```
# %%
#MODEL BUILDING
#Logistic Regression
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression()
lr.fit(x_train,y_train)
```

```
# %%
#accuracy
y_pred = lr.predict(x_test)
log_reg = accuracy_score(y_test,y_pred)
```

```
print(classification_report(y_test, y_pred))
print(f"{round(log_reg*100,2)}% Accurate")

# %%
#Random Forest Classifier
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier()
rf.fit(x_train,y_train)

# %%
#accuracy
y_pred2 = rf.predict(x_test)
ran_for = accuracy_score(y_test,y_pred2)

print(classification_report(y_test, y_pred2))
print(f"{round(ran_for*100,2)}% Accurate")

# %%
#Decision Tree Classifier
from sklearn.tree import DecisionTreeClassifier
dt = DecisionTreeClassifier()
dt.fit(x_train,y_train)

# %%
#accuracy
y_pred3 = dt.predict(x_test)
des_class = accuracy_score(y_test,y_pred3)

print(classification_report(y_test, y_pred3))
print(f"{round(des_class*100,2)}% Accurate")

# %%
#K Neighbors Classifier
from sklearn.neighbors import KNeighborsClassifier
kc = KNeighborsClassifier()
```

```
kc.fit(x_train,y_train)

# %%
#accuracy
y_pred4 = kc.predict(x_test)
kn_class = accuracy_score(y_test,y_pred4)

print(classification_report(y_test, y_pred4))
print(f"{round(kn_class*100,2)}% Accurate")

# %%
#Support Vector Machine
from sklearn import svm
sv = svm.SVC()
sv.fit(x_train,y_train)

# %%
#accuracy
y_pred5 = sv.predict(x_test)
supp_vec = accuracy_score(y_test,y_pred5)

print(classification_report(y_test, y_pred5))
print(f"{round(supp_vec*100,2)}% Accurate")

# %%
score = [log_reg,ran_for,des_class,kn_class,supp_vec]
Models = pd.DataFrame({
    'Classfication Algorithms': ["Logistic Regression","Random Forest
Classifier","Decision Tree Classifier", "K Neighbors Classifier","Support
Vector Machine"],
    'Accuracy': score})
Models.sort_values(by='Accuracy', ascending=False)

# %%
#Random Forest Classifier - highest accuracy
#saving the model
```

```
import pickle
pickle.dump(rf,open('phishing_website.pkl','wb'))
```

GitHub Link: https://github.com/IBM-EPBL/IBM-Project-23105-1659867293

Project Demo Link: Demo