WEB PHISHING DETECTION

A PROJECT REPORT

Submitted by

P.ASHWIN (961219104008)

N.NAVA NISHANTH (961219104301)

T.ESAKKIMUTHU (961219104015)

E.MAHARAJAMUTHU (961219104027)

In partial fulfillment for the award of the degree

Of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

MACHINE LEARNING

WEB PHISHING DETECTION

TEAM ID	PNT2002TMID51508
PROJECT TITTLE	WEBPHISHINGDETECTION
TEAM MEMBERS	**TEAM LEADER :** P.ASHWIN **TEAM MEMBER :** N.NAVA NISHANTH **TEAM MEMBER :** E.MAHARAJA MUTHU **TEAM MEMBER :** T.ESAKKIMUTHU

TABLE OF CONTENT

CHAPTER 1

1.INTRODUCTION

1.1 Project overview

Phishing emails are a routine occurrence for anyone with email in the Internet age. Masking themselves as reputable companies such as using devious means such as the use of company logos and standards, malicious actors, again and again, attempt to lure in individuals from a wide range of technical shades. Phishing ranges extensively in sophistication: from mass-produced misspelt requests for overly specific details to sophisticated spear phishing attacks focused on the details of the individual. The ability of phishing attacks to innocuously harvest your private credentials can leave you mercilessly exposed in our data-intensive world. All it takes is for a user to make the critical mistake of clicking on a single malicious link. Through a simple mistake, a user exposes themselves and their data from anything from drive-by downloads, cross-site scripting attacks to the harvesting of their details in an innocuous web form. Phishing has two main delivery vehicles: emails and websites. Emails being the foremost of these, are most classically associated with phishing. These often include malicious URLs to direct users towards maliciously crafted content. By obfuscating the real destination of a URL through a few simple manipulations, users can quickly find themselves on unknown and insecure ground. Therefore it is vital to tackle this massive worldwide problem. In the United Kingdom (UK) alone, phishing is expected to cost the UK economy as much as £280 million per year. This is encouraging companies such as Google to look into the future of Uniform Resource Locators (URLs) themselves. To tackle the problem of phishing, my project has been focused on tackling the malicious URLs included in them as ‒more than 75% of phishing mails include malicious URLs to phishing sites‖. Existing techniques to handle URLs involve automated phishing detecting (mainly employing machine learning techniques), user training (the best results of which are gained from embedded training) and automated security indicators (providing information to help the users decide). I aim to create a system which incorporates aspects of these techniques, to inform and protect users from malicious.

1.2   Purpose

To solve this problem, I have been building a user-focused tool which seeks to catch problematic URLs before they reach their full malicious potential. The tool that I have developed is a Phishing Learning and Detection tool, built for the Chrome platform in the form of an extension called Catch-Phish. It classifies URLs into one of three safety states using a combination of natural language processing and knowledge acquisition, before providing users with the necessary information to understand how this classification was derived. It helps to train them in URL safety by presenting this information at critical points of intervention. The primary motivation behind this project is the lack of tools which purposefully prevent users from visiting malicious URLs and more crucially inform them of why they have been prevented. This is important due to the significant average availability time of phishing links, which according to Canova et al. [11], was 32 hours and 32 minutes in the first half of 2014. Machine learning techniques are very successful in matching established patterns of URLs but not as successful at identifying new URL variations, which means malicious URLs can go sometime before being detected. The lack of user knowledge of phishing also encourages users to ignore warnings when presented to them. In the UK for example, only 72% of technology users had heard of phishing as a term despite 95% of organizations saying that they train end users [60]. For these reasons, it is important to train users to detected phishing themselves. As the first year of my Mini project, the focus this year was working on both designing the tool and implementing and evaluating the means for how the user would interact with the tool. The project has involved a welcome and generous amount of advice from a PhD student who is an expert in phishing, computer security and URLs and is working on a similar project. This student has produced a lot of the research referenced in this report. However, I make a clear distinction between the work done by myself and this student throughout the report. 1.2.1 Results and Accomplishment

URLs of benign websites were collected from www.alexa.com and The URLs of phishing websites were collected from www.phishtank.com. The data set consists of total 36,711 URLs which include 17058 benign URLs and 19653 phishing URLs. Benign URLs are labeled as ‒0‖ and phishing URLs are labeled.

CHAPTER 2

LITERATURE SURVEY

2.1 Existing problem

Phishing detection schemes which detect phishing on the server side are better than phishing prevention strategies and user training systems. These systems can be used either via a web browser on the client or through specific host-site software presents the classification of Phishing detection approaches. Heuristic and ML based approach is based on supervised and unsupervised learning techniques. It requires features or labels for learning an environment to make a prediction. Proactive phishing URL detection is similar to ML approach. However, URLs are processed and support a system to predict a URL as a legitimate or malicious. Blacklist and Whitelist approaches are the traditional methods to identify the phishing sites. The exponential growth of web domains reduces the performance of the traditional method.
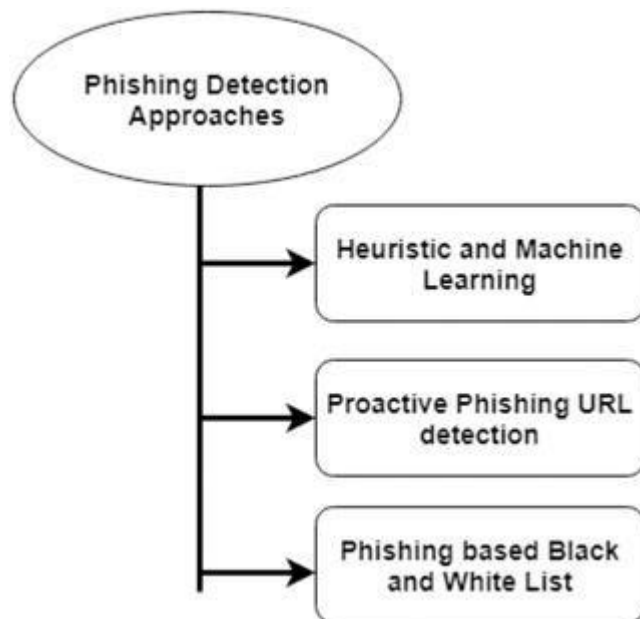
The existing methods rely on new internet users to a minimum. Once they identify phishing website, the site is not accessible, or the user is informed of the probability that the website is not genuine. This approach requires minimum user training and requires no modifications to existing website authentication systems. The performance of the detection systems is calculated according to the following:

1. Number of True Positives (TP): The total number of malicious websites.

2. Number of True Negatives (TN): The total number of legitimate websites.

3. Number of False Positives (FP): The total number of incorrect predictions of legitimate websites as a malicious website.

4. Number of False Negatives (FN): The total number of incorrect predictions of malicious websites as a legitimate website.

Using some benchmark dataset, the accuracy of phishing detection systems is usually evaluated. The familiar phishing dataset to train the ML based techniques are as follows:

Normal dataset:

Alexa Rank is used as a benign and natural website benchmarking dataset. Alexa is a commercial enterprise which carries out web data analysis. It obtains the browsing habits of users from different sources and analyses them objectively for the reporting and classification of Internet web-based URLs. Researchers use the rankings provided by Alexa to collect a number of high standard websites as the normal dataset to test and classify websites. Alexa presents the dataset in the form of a raw text file where each line in the order ascended mentions the grade and domain name of a website.

Phishing dataset:

Phish tank is a familiar phishing website benchmark dataset which is available at https://phishtank.org/. It is a group framework that tracks websites for phishing sites. Various users and third parties send alleged phishing sites that are ultimately selected as legitimate site by a number of users. Thus, Phish tank offers a phishing website dataset in real-time. Researchers to establish data collection for testing and detection of Phishing websites use Phish tank's website. Phish tank dataset is available in the Comma Separated Value (CSV) format, with descriptions of a specific phrase used in every line of the file. The site provides details include ID,URL, time of submission, checked status, online status and target URLs.

2.2 References

1. Anti-Phishing Working Group
   (APWG), https://docs.apwg.org//reports/apwg_trends_report_q4_2019.pdf

2. Jain A.K., Gupta B.B. ‖PHISH-SAFE: URL Features-Based Phishing Detection System Using Machine Learning‖, *Cyber Security. Advances in*

Intelligent *Systems and Computing*, vol. 729, 2018,View Article, Google Scholar

3. Purbay M., Kumar D, ‑Split Behavior of Supervised Machine Learning Algorithms for Phishing URL Detection‖, *Lecture Notes in Electrical Engineering*, vol. 683, 2021,View Article, Google Scholar

4. Gandotra E., Gupta D, ‑An Efficient Approach for Phishing Detection using Machine Learning‖, *Algorithms for Intelligent Systems*, Springer, Singapore, 2021, https://doi.org/10.1007/978-981-15-8711-5_12.

5. Hung Le, Quang Pham, Doyen Sahoo, and Steven C.H. Hoi, ‑URLNet: Learning a URL Representation with Deep Learning for Malicious URL Detection‖, *Conference'17*, Washington, DC, USA, arXiv:1802.03162, July 2017.

6. Hong J., Kim T., Liu J., Park N., Kim SW, ‑Phishing URL Detection with Lexical Features and Blacklisted Domains‖, *Autonomous Secure Cyber Systems*. Springer, https://doi.org/10.1007/978-3-030-33432-1_12.

7. Kumar, A. Santhanavijayan, B. Janet, B. Rajendran and B. S. Bindhumadhava, ‑Phishing Website Classification and Detection Using Machine Learning,‖ *2020 International Conference on Computer Communication and Informatics (ICCCI)*, Coimbatore, India, 2020, pp. 1–6, 10.1109/ICCCI48352.2020.9104161.

8. Hassan Y.A. and Abdelfettah B, ‑Using case- based reasoning for phishing detection", *Procedia Computer Science*, vol. 109, 2017, pp. 281–288., View Article ,Google Scholar

9. Rao RS, Pais AR. Jail-Phish: An improved search engine based phishing detection system. Computers & Security. 2019 Jun 1;83:246–67.

10. Aljofey A, Jiang Q, Qu Q, Huang M, Niyigena JP. An effective phishing detection model based on character level convolutional neural network from URL. Electronics. 2020 Sep;9(9):1514.

11. AlEroud A, Karabatis G. Bypassing detection of URL-based phishing attacks using generative adversarial deep neural networks. In: Proceedings of the Sixth International Workshop on Security and Privacy Analytics 2020 Mar 16 (pp. 53–60).

12. Gupta D, Rani R, ‖Improving malware detection using big data and ensemble learning‖, *Computer Electronic Engineering*, vol. 86, no.106729, 2020.

13. J. Anirudha and P. Tanuja,‖Phishing Attack Detection using Feature Selection Techniques ‖, *Proceedings of International Conference on Communication and Information Processing (ICCIP)*, 2019, http://dx.doi.org/10.2139/ssrn.3418542

14. Wu CY, Kuo CC, Yang CS,‖ A phishing detection system based on machine learning‖ *In: 2019 International Conference on Intelligent Computing and its Emerging Applications (ICEA)*, pp 28–32, 2019.

15. Chiew KL, Chang EH, Tiong WK,‖Utilisation of website logo for phishing detection‖, *Computer Security*, pp.16–26, 2015.

2.3 Problem Statement Definition

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

Phishing sites are malicious websites that imitate as legitimate websites or web pages and aim to steal user's personal credentials like user id, password, and financial information. Spotting these phishing websites is typically a challenging task because phishing is mainly a semantics-based attack, that mainly focus on human vulnerabilities, not the network or software vulnerabilities. Phishing can be elaborated as the process of charming users in order to gain their personal credentials like user-id's and passwords. In this paper, we come up with an intelligent system that can spot the phishing sites. This intelligent system is based on a machine learning model. Our aim through this paper is to stalk a better performance classifier by examining the features of the phishing site and choose appropriate combination of systems for the training of the classifier. Phishing is a non-ethical method comprising both social engineering and technical tricks to capture user's information and sensitive credentials like financial credentials. Some of the social engineering techniques use spam mails, pretending as a legitimate company or organization, that are specially designed to forefront users to knock-off websites that moreover recipients to fall into the trap which steal financial credentials like user-ids and passwords.

CHAPTER-3

IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

An **empathy map** is a collaborative visualization used to articulate what we know about a particular type of user. It externalizes knowledge about users in order to 1) create a shared understanding of user needs, and 2) aid in decision making. Traditional empathy maps are split into 4 quadrants (*Says*, *Thinks*, *Does*, and *Feels*), with the user or persona in the middle. Empathy maps provide a glance into who a user is as a whole and are not chronological or sequential. Empathy maps should be used throughout any UX process to establish common ground among team members and to understand and prioritize user needs. In user-centered design, empathy maps are best used from the very beginning of the design process**.**
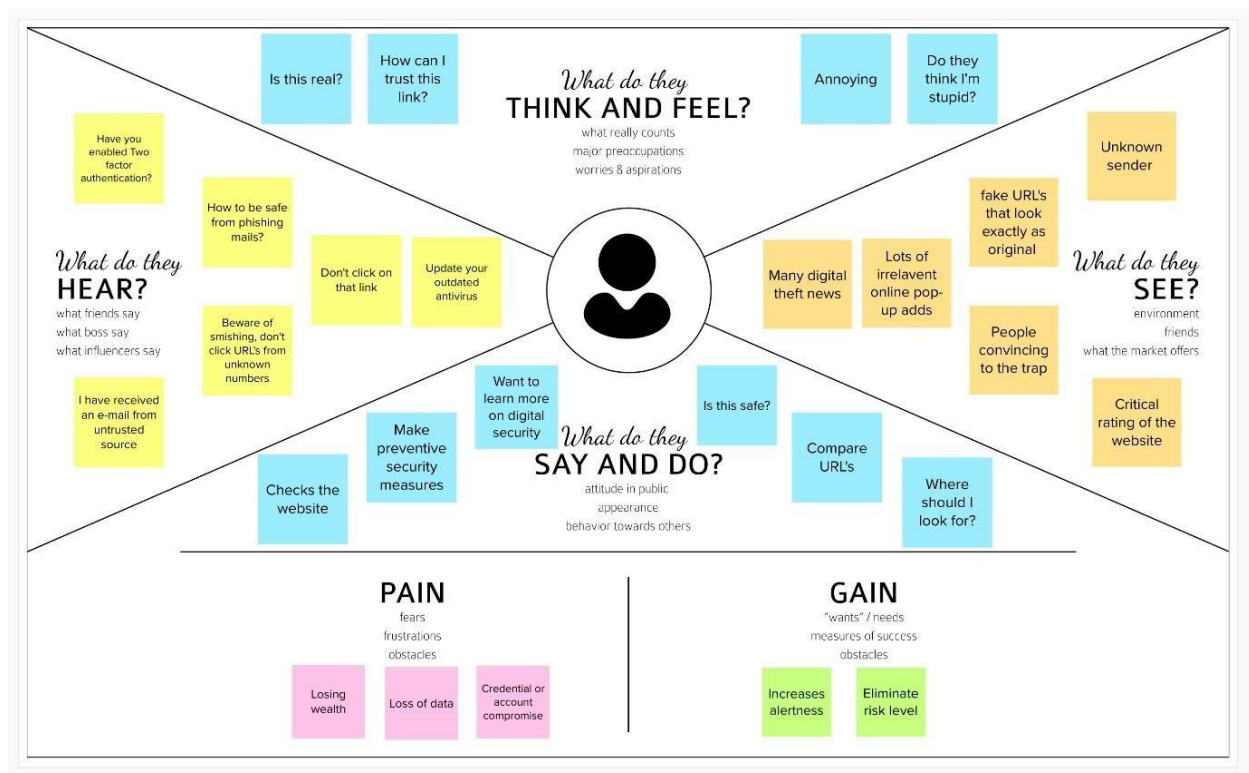
3.2 Ideation & Brainstorming

Ideation is often closely related to the practice of brainstorming, a specific technique that is utilized to generate new ideas. 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. Brainstorming is usually conducted by getting a group of people together to come up with either general new ideas or ideas for solving a specific problem or dealing with a specific situation. For example, a major corporation that recently learned it is the object of a major lawsuit may want to gather together top executives for a brainstorming session on how to publicly respond to the lawsuit being filed.

Brainstrom & Idea Prioritization :

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

The lines between ideation and brainstorming have become a bit more blurred with the development of several brainstorming software programs, such as Bright idea and Idea wake. These software programs are designed to encourage employees of companies to generate new ideas for improving the companies' operations and, ultimately, bottom-line profitability. The programs often combine the processes of ideation and brainstorming in that individual employees can use them, but companies may simulate brainstorming sessions by having several employees all utilize the software to generate new ideas intended to address a specific purpose.

Increase email security

Restrict access to sensitive data

ensure for updates

validating DNS

Don't ignore those updates

Delete the email without opening it.

Double-check the domain

Don't give any one person unilateral authority to approve and send electronic payments

Use two-factor authentication

making of redirecting the url

extracting the info of some links

Use your browsers phishing list

Counting length of URL

Check for fake IP address

Identify fake website

❤
Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

Knowing current updates

Identify phishing email

Save frequently reopening pages

Diagnose if there is any Virus implemented

Participants in a brainstorming session are encouraged to freely toss out whatever ideas may occur to them. The thinking is that by generating a large number of ideas, the brainstorming group is likely to come up with a suitable solution for whatever issue they are addressing

3.3 Proposed Solution

Having hooked your audience into the problem, now you want to paint a picture of what the world will be like when you solve the problem. Your proposed solution should relate the current situation to a desired result and describe the benefits that will accrue when the desired result is achieved. So, begin your proposed solution by briefly describing this desired result.

Note: now that you have described the final result, think of your iterative enhancement plan as showing your audience the steps by which you will lead them to that result.

Define CS, fit into CC | **Explore AS, differentiate**

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

---

Focus on J&P, tap into BE, understand RC | **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.

Identify strong TR & EM | **Extract online & offline CH of BE**

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 #7

- 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.

3.4 Problem Solution fit

The Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realized for it actually solves the customer's problem.

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) *CS*

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

6. CUSTOMER CONSTRAINTS *CC*

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

5. AVAILABLE SOLUTIONS *AS*

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

Explore AS, differentiate

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

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

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

9. PROBLEM ROOT CAUSE *RC*

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

7. BEHAVIOUR *BE*

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

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

IDENTIFY STRONG TR & EM

3. TRIGGERS *TR*

- Hearing cyber-attack crimes in news
- Being redirected to unwanted links often

10. YOUR SOLUTION *SL*

- In ML, Decision tree classifier used to detect URL
- Database of URL maintained as whitelist or blacklist
- Use data mining algorithm to detect

8. CHANNELS of BEHAVIOUR *CH*

8.1 Online
   Install anti phishing software

8.2 Offline
   Install firewall

4. EMOTIONS: BEFORE / AFTER *EM*

Before: They lose their login credentials and feel insecure.

After: They feel confident when they receive alerts about phishing websites.

IDENTIFY STRONG TR & EM

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 Functional requirement

Functional requirements are the desired operations of a program, or system as defined in software development and systems engineering. The systems in systems engineering can be either software electronic hardware or combination software-driven electronic

- Address Bar based Features.
- Abnormal Based Features
- HTML and JavaScript Based Features.
- Domain Based Features.

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Verifying input	User inputs an URL in necessary field to check its validation
FR-2	User Confirmation	Confirmation through email Confirmation via OTP
FR-3	Authentication	Confirmation of google firebase
FR-4	User Permission	User must give permission access to the search engine
FR-5	Website Evaluation	Model checks for the website in appearance of whitelist and blacklist
FR-6	Real time monitoring	The use of extension plugin should provide a warning popup when they visit a website that is phished. Extension plugin will have the capability to also detect latest and new phishing website

4.2 Non-Functional requirements

Nonfunctional 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.

Non-functional Requirements:

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

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	This is very user friendly any people with less knowledge also can easily understanded that they are using the fake websites through out alert message
NFR-2	Security	This website is secured as one cant hack our detection website , our website is easily trusted ones and they will be saved user financial information loss
NFR-3	Reliability	It has good consistency and performs as it actively detect the fake website and protect the confidential information and financial loss of the user
NFR-4	Performance	Web phishing detection performs high. It is very efficient, very easy to understand and it has a high security and scalability

| NFR-5 | Availability | This detection website is available at any system like smart phones , laptop , smart watch, desktop and other electronic devices. User can easily got it |
| NFR-6 | Scalability | The total execution time of our approach in phishing web page detection is around 2-3sec,which is acceptable to our environment .As input size and execution time is increase it make the system difficult to handle and increase the stress |

CHAPTER 5
PROJECT DESIGN

5.1 Data Flow Diagrams

 A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyze an existing system or model a new one. Like all the best diagrams and charts, a DFD can often visually ‒say‖ things that would be hard to explain in words, and they work for both technical and nontechnical audiences, from developer to CEO. That's why DFDs remain so popular after all these years.

WEB PHISHING DETECTION

WEBSITES

Feature Extraction

Creation of training and testing dataset

- Decision tree classifier using ML
- IBM Watson

Performance evaluation metrices

Result

Legitimate website

Phishing website
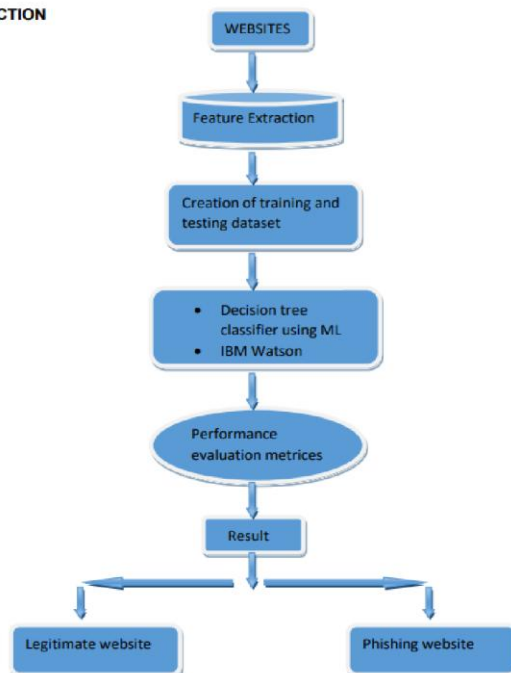
5.2 Solution & Technical Architecture

This project will be approached based on decision tree, random forest and SVM algorithms.

- The data is collected from the dataset and it is pre-processed.
- Then it is applied to the machine learning algorithmic model and analyzed.
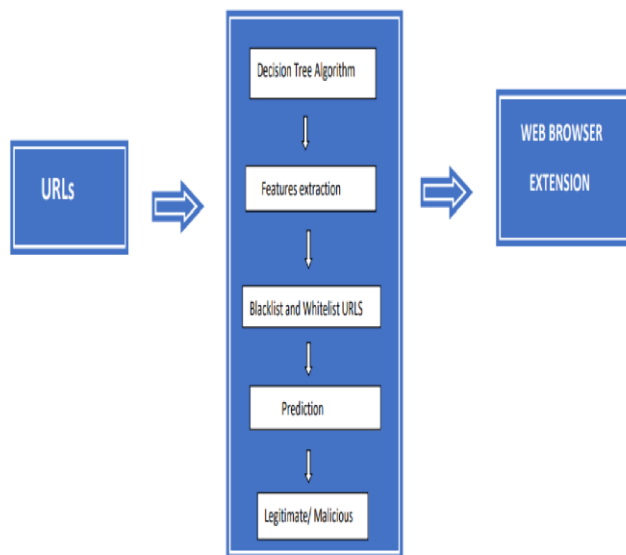- The model is trained and tested according to the problem specified.
- Python Flask and IBM cloud facility is used in developing this project.

Technical Architecture:

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

WEB PHISHING DETECTION:

S No	Component	Description	Technology
1.	User Interface	User's interacts with application as a website	HTML, CSS
2.	prediction of phishing website	Detecting the phishing websites in the dataset	Python-Flask
3.	Dataset	Provided by the IBM team	IBM Watson STT service
4.	Storage	Store the contents of the web application	IBM Watson Assistant
5.	Cloud Database	Database Service on Cloud	IBM cloud.
6.	Machine Learning Model	Purpose of Machine Learning Model	Decision tree, Random Forest.
7.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration Cloud Server Configuration	Local, IBM Cloud .

5.3 User Stories

A user story is the smallest unit of work in an agile framework. It's an end goal, not a feature, expressed from the software user's perspective .A user story is an informal, general explanation of a software feature written from the perspective of the end user or customer. The purpose of a user story is to articulate how a piece of work will deliver a particular value back to the customer. Note that "customers" don't have to be external end users in the traditional sense, they can also be internal customers or colleagues within your organization who depend on your team.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can register & access the dashboard with Gmail Login	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can access my dashboard	High	Sprint-1
	Dashboard	USN-6	As a user, I can access the dashboard to get information	I can access my application	High	Sprint-1
Customer (Web user)	Registration	USN-7	As a web user, I can register my details in official websites and I will create strong passwords	I can access my dashboard/account safely	High	Sprint-1
	Login & Dashboard		As a web user, I can login into application by using my user id and password	I can access the resources	High	Sprint-1
Customer Care Executive	Login	CCE-1	As a CCE I can login to website using user id and password and I can interact with the user	I can access the website	High	Sprint-1
	Dashboard	CCE-2	As a CCE I can login to dashboard using user id and password and I can interact with the user and I can explain the app usage and rectify their issues.	I can access the resources	High	Sprint-1
Administrator	Login & Dashboard	A-1	As an administrator, I can access the dashboard and direct activities.	I will maintain the database safely	High	Sprint-1

List-based phishing detection methods use either whitelist or blacklist-based technique. A blacklist contains a list of suspicious domains, URLs, and IP addresses, which are used to validate if a' URL is fraudulent.

CHAPTER 6
PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

During sprint planning, we break the stories down into tasks, estimate those tasks, and compare the task estimates against our capacity. It's that, not points, that keep us from overcommitting in this sprint. No need to change the estimate.

Estimation is an essential management operation used to determine an approximate time frame required to start and finish any process in a controlled environment. It's crucial to any project planning to not go past the time limits, set budgets, and available resources.

| 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.	2	High	Vimala T
Sprint-1	Authentication	USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Sindhuja M
Sprint-1	Analysis of websites	USN-3	Use whitelist and blacklist analysis to analyse the website	2	Low	Sabarmathi S
Sprint-2	Storage	USN-4	Store the backlisted websites In a specified database allotted for it using IBM cloud.	2	Medium	Fathima Iffadha SM
Spirint-2	Decide ML model	USN-5	Selecting the best model and proceed further.	3	High	Sabarmathi.S
Spirint-3	Creation	USN-6	Creation of the application and hosting it in IBM cloud.	5	High	Vimala.T Sindhuja.M Sabarmathi.S Fathima Iffadha SM
Spirint-3	Result display	USN-7	Display whether the website is legitimate or illegitimate.	4	Medium	Sindhuja.M
Sprint-4	Login	USN-8	As a user, I can log into the application by entering email & password	1	High	Fathima Iffadha SM
Spirint-4	Dashboard	USN-9	Display the list of blacklisted websites that are identified.	2	Low	Sabarmathi S

For super-fast Agile estimation, the items to be estimated are simply placed by the group in one of three categories: big, uncertain and small. The group starts by discussing a few together, and then, like the Bucket System, uses divide-and-conquer to go through the rest of the items.

6.2 Sprint Delivery Schedule

sprint is a set period of time during which specific work has to be completed and made ready for review. Each sprint begins with a planning meeting. During the meeting, the product owner (the person requesting the work) and the development team agree upon exactly what work will be accomplished during the sprint. The development team has the final say when it comes to determining how much work can realistically be accomplished during the sprint, and the product owner has the final say on what criteria need to be met for the work to be approved and accepted.

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	18 Nov 2022

The deliverables of a sprint aren't as predictable as they are for other projects. Sprint participants have produced sketches and drawings, writing, photographs, comic strips, videos and fully coded working prototypes. The answer is whatever's right to answer the problem. The Scrum Developer is the professional responsible for creating the project deliverables, together with the rest of the Scrum team. As described in the Scrum Guide, there are three core roles in Scrum, responsible for meeting the project objectives: the Product Owner, the Scrum Master and the Development Team.Grooming the backlog and deciding which work to complete in the upcoming sprint.Agile delivery is an iterative approach to software delivery in which teams build software incrementally at the beginning of a project rather than ship it at once upon completion. Agile development means taking iterative, incremental, and lean approaches to streamline and accelerate the delivery of projects. the Scrum Master is not responsible for delivery. Not directly. The Scrum Master is part of the Scrum Team. The Scrum Team is responsible for delivery of working software.
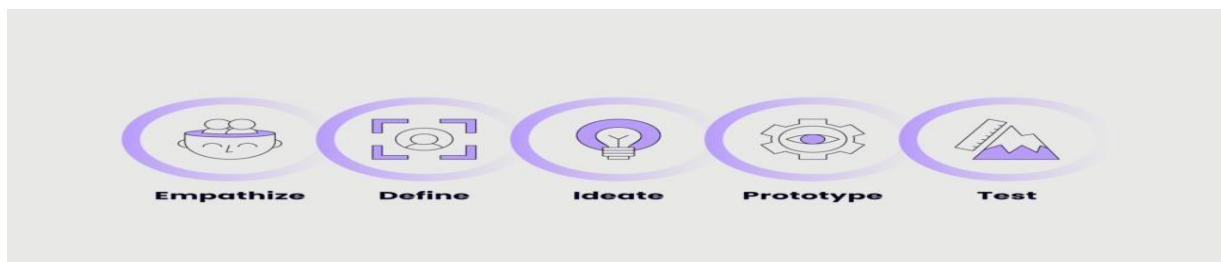
A burn down chart shows the amount of work that has been completed in an epic or sprint, and the total work remaining. Burn down charts are used to predict your team's likelihood of completing their work in the time available. They're also great for keeping the team aware of any scope creep that occurs.
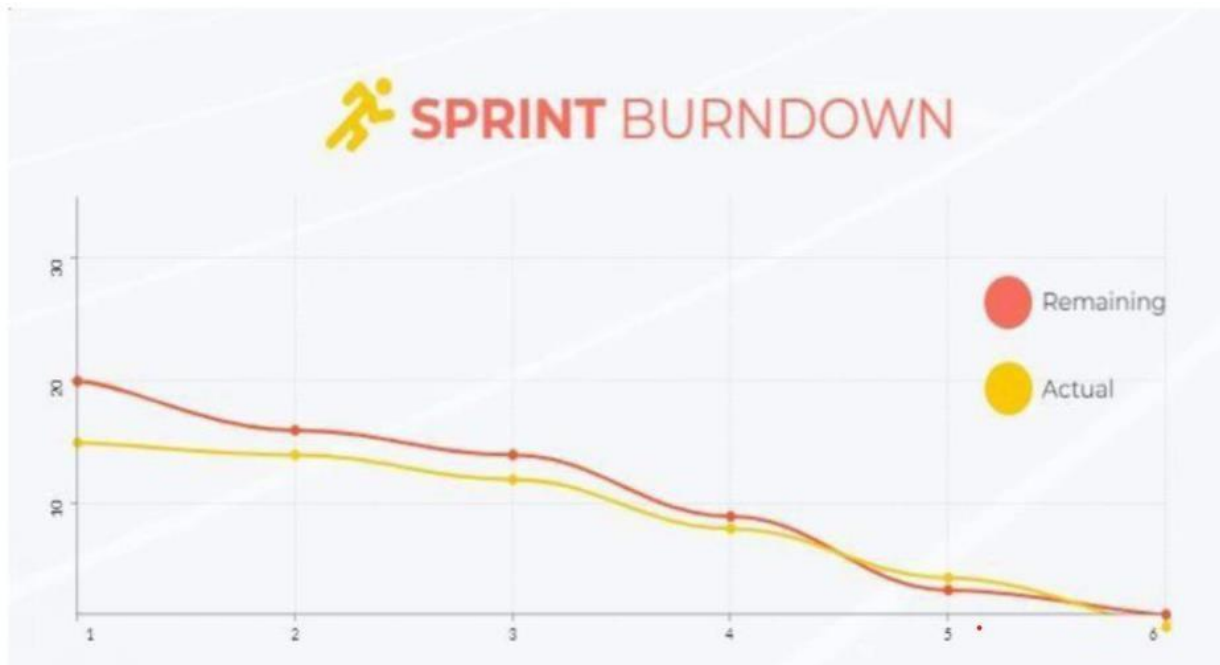

```python
import requests

# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = "cWGD5yTjEpEGtqPpvHPDBElN5eXFS7eh2JRDyUWhySMW"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey":
 API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}

# NOTE: manually define and pass the array(s) of values to be scored in the next line
payload_scoring = {"input_data": [{"field": [["UsingIP","LongURL","ShortURL","Symbol@","Redirecting//","PrefixSuffix-","SubDomains","HTTPS","DomainReg
]], "values": [[1,1,1,1,1,-1,-1,-1,-1,1,1,1,1,-1,-1,1,1,1,0,1,1,1,1,-1,-1,-1,-1,1,0,1]]}]}

response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/084b5c52-f617-40ef-a0e8-3e6cf79ae447/predictions?version=2022-11
 headers={'Authorization': 'Bearer ' + mltoken})
print("Scoring response")
predictions=response_scoring.json()
#print(predictions)
pred=print(predictions['predictions'][0]['values'][0][0])
if(pred != 1):
    print("The Website is secure.. Continue")
else:
    print("The Website is not Legitimate... BEWARE!!")
```

a sprint schedule is a document that outlines sprint planning from end to end. It's one of the first steps in the agile sprint planning process—and something that requires adequate research, planning, and communication.

6.3 Report from JIRA

When JIRA sends either standard notifications or user invitations to a mail server, they are listed as phishing attempts rather than legitimate emails

The mail server is receiving a constant stream of concurrent multiple email notifications from the same sender which, in turn, triggers security measures on the server which handle these messages as phishing attempts.

Resolution

- Check the base url of JIRA to see if it is set as a direct ip address with port number.
- Example: **http://10.10.10.10:8080**
- Some email servers (such as Microsoft Outlook) will consider messages from non-DNS urls as phishing attempts. You can correct this behaviour by setting JIRA's base url to a url address such as **http://my-jira.com.**
- Sometimes when certain mail servers receive multiple emails from the same sender security measures are triggered that will then list those emails as phishing messages. For this, it is best to check with the local mail server administrator for further assistance and confirmation.

Pages / Demo Project Home   🔓   ✏ Edit   ☆ Save for later   👁 Watching   ↗ Share   ⋯

Open issues in JIRA

Created by Atlassian OnDemand [Administrator], last modified 3 minutes ago

| Key	Summary	T	Created	Updated	Due	Assignee	P	Status	Resolution
WTP-10	Back end	⚡	Dec 30, 2016	Dec 30, 2016		Unassigned	↑	TO DO	Unresolved
WTP-9	Front end	⚡	Dec 30, 2016	Dec 30, 2016		Unassigned	↑	TO DO	Unresolved
WTP-8	Investigate vendor options	☑	Dec 22, 2016	Dec 30, 2016		Unassigned	↑	TO DO	Unresolved
WTP-7	Gather user metrics	☑	Dec 22, 2016	Dec 30, 2016		Unassigned	↑	TO DO	Unresolved
WTP-6	Website's color scheme is wrong	⬛	Dec 22, 2016	Dec 30, 2016		Unassigned	↑	TO DO	Unresolved
WTP-5	Back button leads to 404 page	⬛	Dec 22, 2016	Dec 30, 2016		Unassigned	↑	TO DO	Unresolved
WTP-4	Create new home page	☑	Dec 22, 2016	Dec 30, 2016		Unassigned	↑	TO DO	Unresolved
WTP-3	As a user, the first page I see is the home page	⬜	Dec 22, 2016	Dec 30, 2016		Jose	↑	TO DO	Unresolved
WTP-2	As an external client, I can navigate to the pages most important to me through my profile	⬜	Dec 22, 2016	Dec 30, 2016		Alana	↑	TO DO	Unresolved
WTP-1	Hide the context menu when user right-clicks on an issue	⬜	Dec 22, 2016	Dec 30, 2016		Will	↑	TO DO	Unresolved

10 issues ↻ Refresh

CHAPTER 7

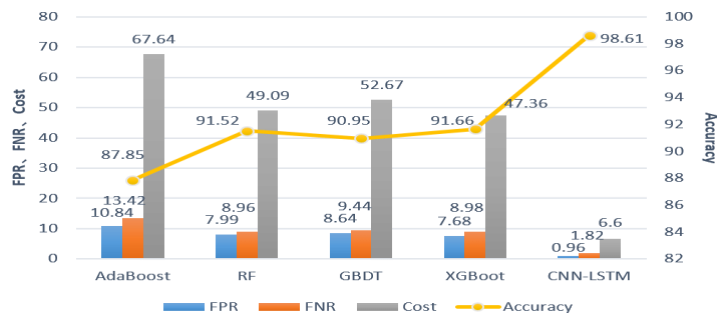# CODING & SOLUTIONING

7.1 Feature 1

- Phishing URL detection

Phishing is a form of fraud in which the attacker tries to learn sensitive information such as login credentials of person in email or other communication channels.
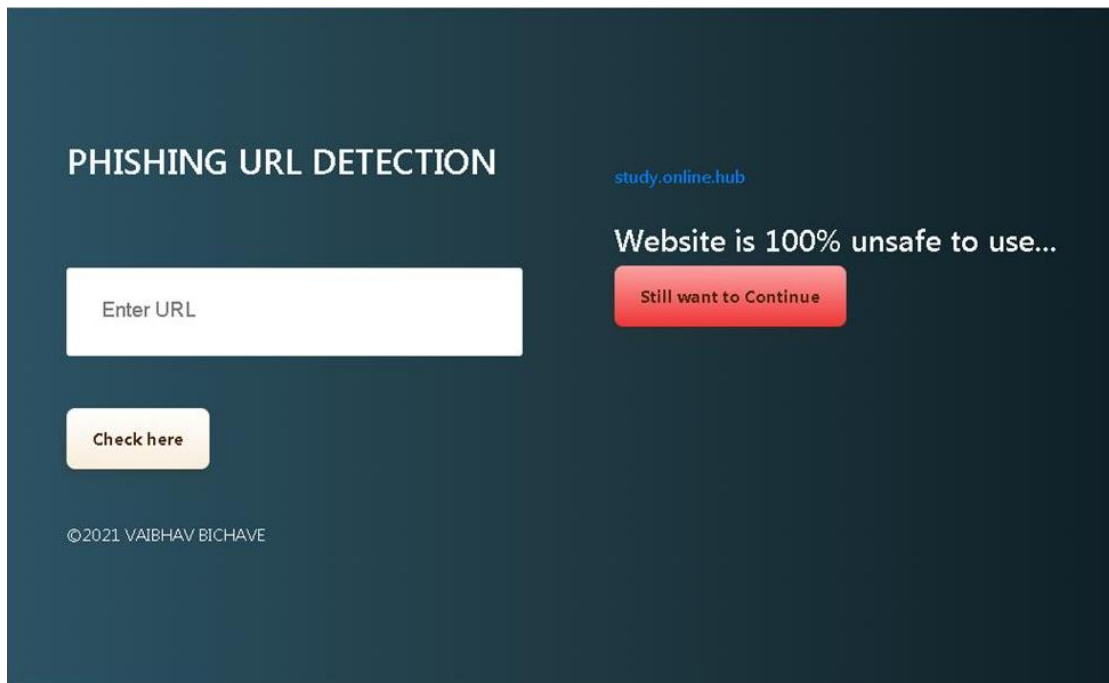

7.2 Feature 2

- Accuracy of an URL

we present a way to detect such malicious URL addresses with almost 100% accuracy using convolutional neural network.

7.3 Work flow

We propose a phishing detection approach, which extracts efficient features from the URL and HTML of the given webpage without relying on third-party services. Thus, it can be adaptable at the client side and specify better privacy. An efficient solution for phishing detection that extracts the features from website's URL and HTML source code proposed.
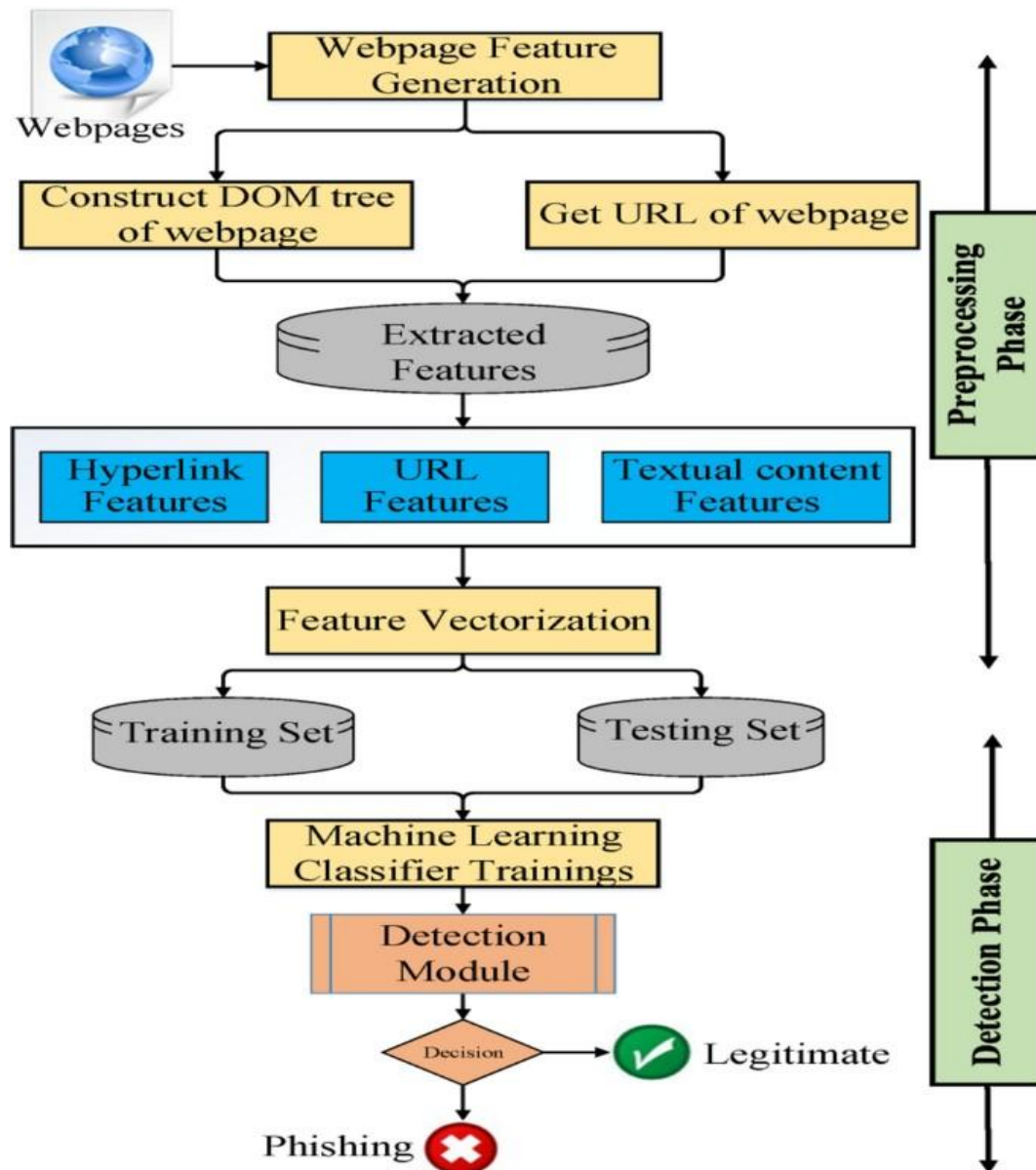
CHAPTER 8

TESTING

8.1 Test cases

For the URL verifier module in the ISOT phishing detection system, phishing detection is done using 16 different heuristic rules. In the system, 11 main classes were defined, and 1 class was defined with 5 sub-classes. This covers all 16 heuristic rules. To test the system, 15 test cases were designed using assertion methods. Ten test cases were designed to test the 10 main classes and 5 test cases were designed to test the class with five sub-classes. The getter-setter method was used to test the class with five sub-classes. The getter method is used to obtain or retrieve a variable value from the class, and the setter method is used to store the variables.

The class with five sub-classes checks the 5 different heuristic rules, length of the URL, number of dots and slashes in the URL, presence of @ symbols in the URL, IP address mentioned in the URL, and the presence of special character such as ',', '_', ';' in the URL. Initially, only a single test case was created for the class with five sub-classes, but it was failing as this class has five methods as shown in Figure 4.11. After applying the getter setter method, all the test cases passed without any issues. The test results are shown in Figure 4.12. assert Not Null() is used to check if the input URL is not empty, and assert Array Equals() is used to compare the result from the detection method with the expected result.
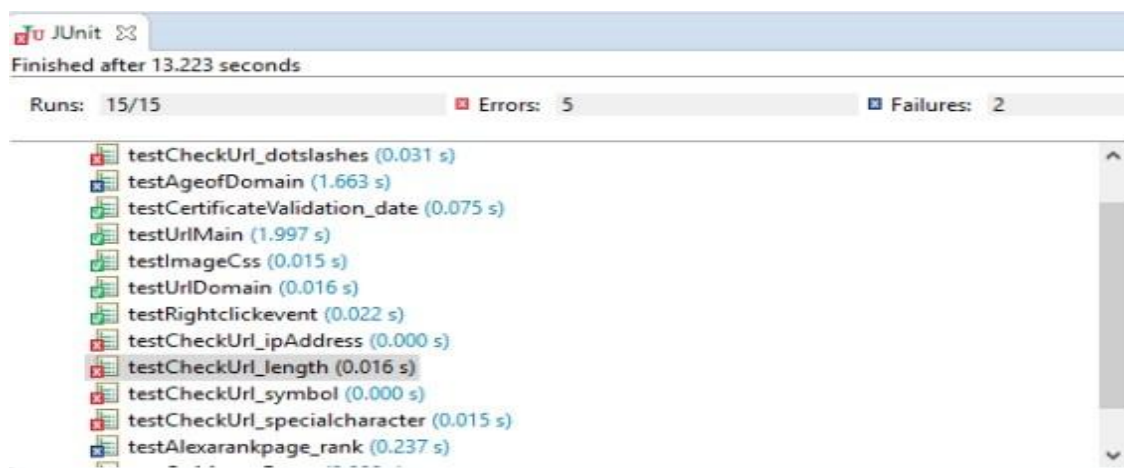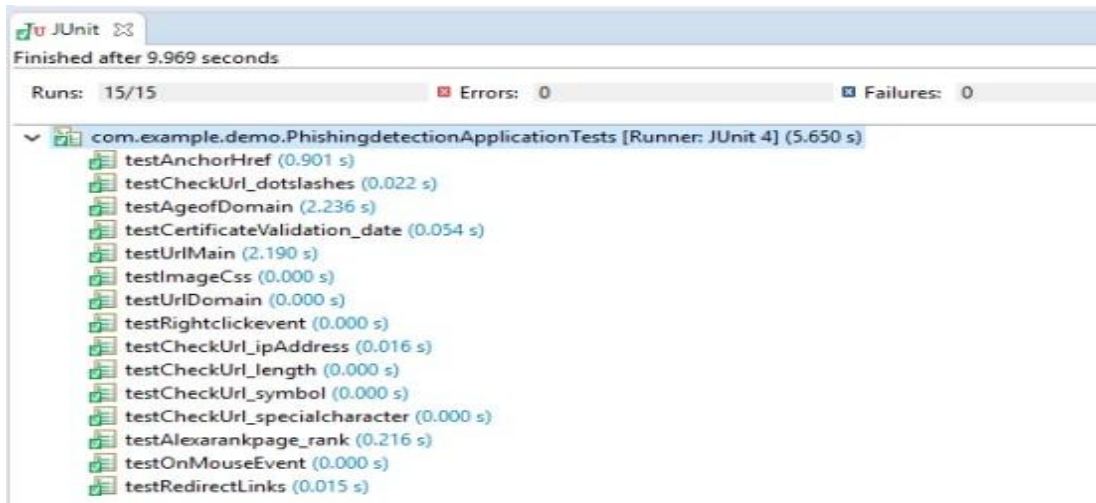

Figure 8.1: Some test cases are failed

Figure 8.2: All test cases are passed.

Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)	BUG ID	Executed By
index.html	Login/Signup popup should display	Working as expected	Pass		Y		Sinduja.M,sabarmathi.S
index.html	Application should show below UI elements: a.url text box b.Check text box c.Continue text box	Working as expected	Pass		Y		Fathima Iffadha.S.M,Vimala.T
Url prediction:https://code-projects.org/	navigate to website	Working as expected	Pass		Y		Sabarmathi.S,Fathima Iffadha.S.M
Url prediction:https://code-projects.org/	Application should show 'Url ' validation message.	Working as expected			Y		Sindhuja.M,Vimala.T

8.2 User Acceptance Testing

Ụser Acceptance Testing (UAT) is a type of testing performed by the end user or the client to verify/accept the software system before moving the software application to the production environment. UAT is done in the final phase of testing after functional, integration and system testing is done.
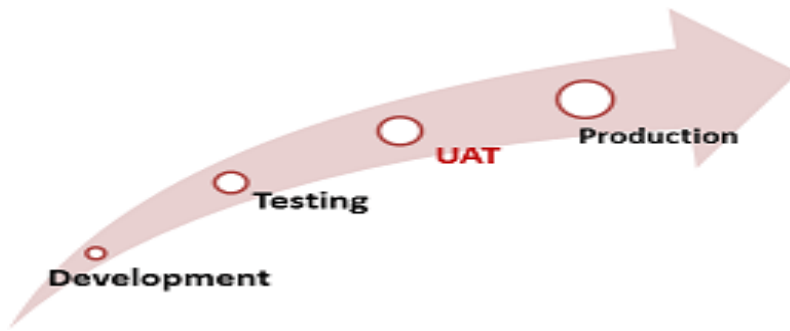

The main **Purpose of UAT** is to validate end to end business flow. It does not focus on cosmetic errors, spelling mistakes or system testing. User Acceptance Testing is carried out in a separate testing environment with production-like data setup. It is kind of black box testing where two or more end-users will be involved.

UAT is performed by –

- Client
- End users

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	5	0	1	4
Client Application	49	0	2	45
Security	8	0	0	7
Outsource Shipping	2	0	0	5
Exception Reporting	11	0	2	9
Final Report Output	5	0	0	6
Version Control	3	0	1	3

UAT PROCESS
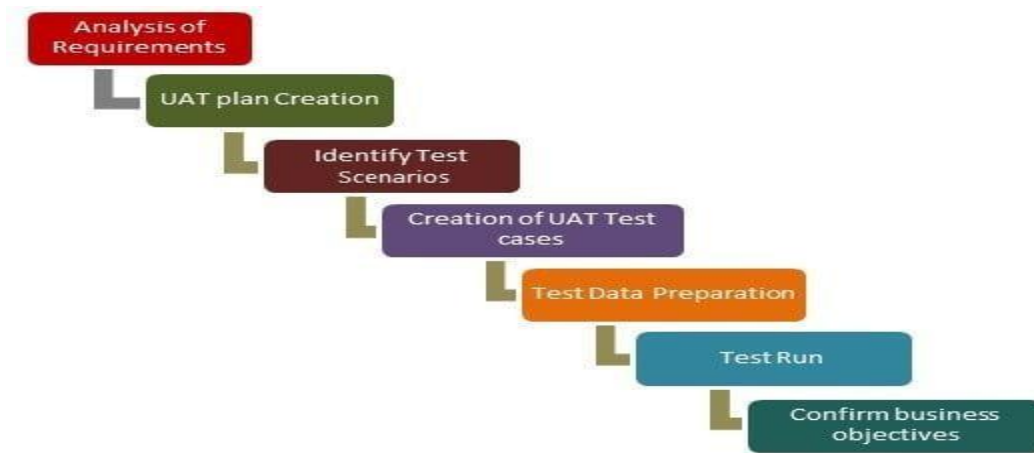
Need of User Acceptance Testing arises once software has undergone Unit, Integration and System testing because developers might have built software based on requirements document by their own understanding and further required changes during development may not be effectively communicated to them, so for testing whether the final product is accepted by client/end-user, user acceptance testing is needed.

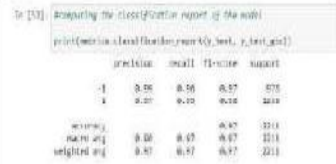| Section	Total Cases	Not Tested	Fail	Pass
Print Engine	5	0	1	4
Client Application	49	0	2	45
Security	8	0	0	7
Outsource Shipping	2	0	0	5
Exception Reporting	11	0	2	9
Final Report Output	5	0	0	6
Version Control	3	0	1	3

CHAPTER 9

RESULT

9.1 Performance Metrics

In a classification problem, the category or classes of data is identified based on training data. The model learns from the given dataset and then classifies the new data into classes or groups based on the training. It predicts class labels as the output, such as Yes or No, 0 or 1, Spam or Not Spam, etc. To evaluate the performance of a classification model, different metrics are used, and some of them are as follows:

- Accuracy
- Confusion Matrix
- Precision
- Recall
- F-Score
- AUC(Area Under the Curve)-ROC

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

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

CLASSIFICATION REPORT:

```
In [52]: #computing the classification report of the model

         print(metrics.classification_report(y_test, y_test_gbc))
```

```
              precision    recall  f1-score   support

          -1       0.99      0.96      0.97       976
           1       0.97      0.99      0.98      1235

    accuracy                           0.97      2211
   macro avg       0.98      0.97      0.97      2211
weighted avg       0.97      0.97      0.97      2211
```
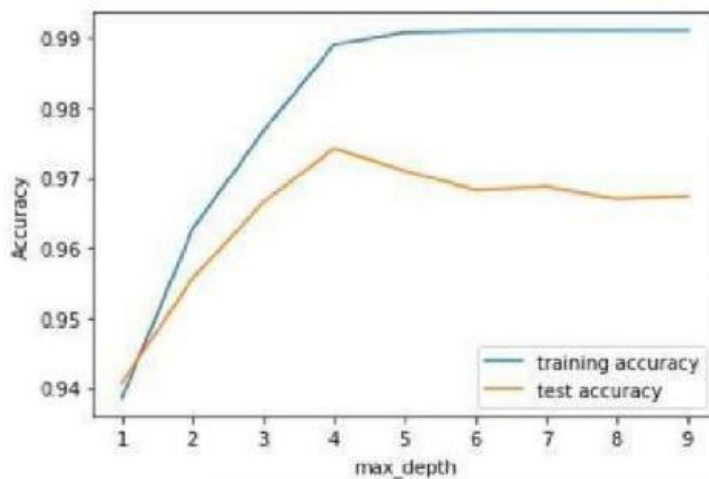

Out[83]:

	ML Model	Accuracy	f1_score	Recall	Precision
0	Gradient Boosting Classifier	0.974	0.977	0.994	0.986
1	CatBoost Classifier	0.972	0.975	0.994	0.989
2	Random Forest	0.969	0.972	0.992	0.991
3	Support Vector Machine	0.964	0.968	0.980	0.965
4	Decision Tree	0.958	0.962	0.991	0.993
5	K-Nearest Neighbors	0.956	0.961	0.991	0.989
6	Logistic Regression	0.934	0.941	0.943	0.927

CHAPTER 10

ADVANTAGES & DISADVANTAGES

Advantages

Blacklists

- Requiring low resources on host machine
- Effective when minimal FP rates are required

Heuristics and visual similarity

- Mitigate zero hour attacks.

Machine Learning

- Mitigate zero-hour attacks.
- Construct own classification models.

Disadvantages

Blacklists

- Mitigation of zero-hour phishing attacks.
- Can result in excessive queries with heavily loaded server.

Heuristics and visual similarity

- Higher FP rate than blacklists.
- High computational cost.

Machine Learning

- Time consuming
- Costly
- Huge number of rules
- loss of money
- loss of intellectual property
- damage to reputation
- disruption of operational activities
- Phishing scams involve sending out emails or texts disguised as legitimate sources.

CHAPTER 11

CONCLUSION

Phishing websites are being designed to trick people into submitting credentials to access private information and assets by making the sites look like legitimate websites. Phishing attacks through URLs embedded in emails or SMS messages are one of the major issues faced by Internet users because of the huge number of online transactions performed daily. Phishing attacks cause losses to organizations, customers and Internet users. In this project, testing utilities were developed to support test automation for different aspects of the ISOT phishing detection system. In particular, two kind of test utilities were developed. 1. A mechanism to populate live test email accounts from different datasets, including spam, phishing, and legitimate emails, allowing online testing of the ISOT phishing detection system. 2. Automated test cases were used to unit test one of the modules of the ISOT phishing detection system. In this report, the data collection process for phishing, spam and legitimate emails was discussed. Further, the ISOT message security system was explained, and the design and implementation of the service to read the eml files from the collected datasets and send them to the test accounts was explained. The testing was done using the JUnit framework to check the functionality of the different heuristic rules of the system. The results of the unit tests were verified using assertion methods in the JUnit framework.

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

CHAPTER 12

FUTURE SCOPE

In the future, optimization can be done in the test units and these units can be made fully automated using Robot-Framework. This is important if more heuristics rules are included in the detection system. If the URL length is very long i.e. more than a million characters, then the system may crash. To prevent this situation, a timeout feature can be added when determining the URL length.

In future if we get structured dataset of phishing we can perform phishing detection much more faster than any other technique. In future we can use a combination of any other two or more classifier to get maximum accuracy. We also plan to explore various phishing techniques that uses Lexical features, Network based features, Content based features, Webpage based features and HTML and JavaScript features of web pages which can improve the performance of the system. In particular, we extract features from URLs and pass it through the various classifiers

CHAPTER 13

APPENDIX

SOURCE CODE

App.py

#importing required libraries

From flask import Flask, request, render _template

import requests

import numpy as np

import warnings

import pickle

warnings.filterwarnings('ignore')

from feature import Feature Extraction

```python
file = open("model.pkl","rb")
gbc = pickle.load(file)
file.close()


API_KEY = "UWEsUaH1i-FABXxbCpQ9lcPk5E0jIaivG8i-veVF9zJj"
token_response =requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey":
 API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]


header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}


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


        url = request.form["url"]
        obj = FeatureExtraction(url)
        x = np.array(obj.getFeaturesList()).reshape(1,30)


        #1 is safe
        #-1 is unsafe
        y_pro_phishing = gbc.predict_proba(x)[0,0]
        y_pro_non_phishing = gbc.predict_proba(x)[0:,1]
        print( y_pro_phishing,y_pro_non_phishing)
        # if(y_pred ==1 ):
        pred = "It is {0:.2f} % safe to go ".format(y_pro_phishing*100)
        payload_scoring                =                {"input_data":                [{"field":
[["UsingIP","LongURL","ShortURL","Symbol@","Redirecting//","PrefixSuffix-
```

","SubDomains","HTTPS","DomainRegLen","Favicon","NonStdPort","HTTPSDomain URL","RequestURL","AnchorURL","LinksInScriptTags","ServerFormHandler","Info Email","AbnormalURL","WebsiteForwarding","StatusBarCust","DisableRightClick"," UsingPopupWindow","IframeRedirection","AgeofDomain","DNSRecording","WebsiteT raffic","PageRank","GoogleIndex","LinksPointingToPage","StatsReport"

```
    ]], "values":obj}]}


    response_scoring                =                requests.post('https://us-
south.ml.cloud.ibm.com/ml/v4/deployments/phishing_1/predictions?version=2022-11-11',
json=payload_scoring,headers={'Authorization': 'Bearer ' + mltoken})

    print("Scoring response")

    predictions=response_scoring.json()

    print(predictions)

    pred=print(predictions['predictions'][0]['values'][0][0])

    if(pred != 1):

        print("The Website is secure.. Continue")

    else:

        print("The Website is not Legitimate... BEWARE!!")


 return render_template('index.html',xx =round(y_pro_non_phishing,2),url=url )

 return render_template("index.html", xx =-1)

if _name_ == "_main_":

   app.run(debug=True)
```

PROJECT DEMO LINK

https://screenapp.io/#/shared/a6fb605b-24ff-49ec-8bcb-30c83aab492d

GITHUB

https://github.com/IBM-EPBL/IBM-Project-10606-1659192520