

PHISHING WEBSITE DETECTION PROJECT REPORT

Team ID: PNT2022TMID21424

Team Members:

Shri Krishnaa V N (917719D092) (Team Lead)

Shankar Mahadevan G (917719D087)

Sanjeev K (917719D077)

Siddharthan S (917719D093)

Manivel Prakash V (917719D124)

CONTENTS

1. INTRODUCTION

- 1.1 Project Overview
- 1.2 Purpose

2. LITERATURE SURVEY

- 2.1 Existing problem
- 2.2 References
- 2.3 Problem Statement Definition

3. IDEATION & PROPOSED SOLUTION

- 3.1 Empathy Map Canvas
- 3.2 Ideation & Brainstorming
- 3.3 Proposed Solution
- 3.4 Problem Solution fit

4. REQUIREMENT ANALYSIS

- 4.1 Functional requirement
- 4.2 Non-Functional requirements

5. PROJECT DESIGN

- 5.1 Data Flow Diagrams
- 5.2 Solution & Technical Architecture
- 5.3 User Stories

6. PROJECT PLANNING & SCHEDULING

- 6.1 Sprint Planning & Estimation
- 6.2 Sprint Delivery Schedule
- 6.3 Reports from JIRA

7. CODING & SOLUTIONING (Explain the features added in the project along with code)

- 7.1 Classification of Website URLs
- 7.2 ML Web Deployment
- 7.3 Website – User Interface

8. TESTING

8.1 Test Cases

8.2 User Acceptance Testing

9. RESULTS

9.1 Performance Metrics

10. ADVANTAGES & DISADVANTAGES

11. CONCLUSION

12. FUTURE SCOPE

13. APPENDIX

Source Code

GitHub & Project Demo Link

1.INTRODUCTION

1.1 Project Overview :

As we have moved most of our financial, work related and other daily activities to the internet, we are exposed to greater risks in the form of cybercrimes. URL based phishing attacks are one of the most common threats to the internet users. In this type of attack, the attacker exploits the human vulnerability rather than software flaws. It targets both individuals and organizations, induces them to click on URLs that look secure, and steal confidential information or inject malware on our system. Different machine learning algorithms are being used for the detection of phishing URLs, that is, to classify a URL as phishing or legitimate. Researchers are constantly trying to improve the performance of existing models and increase their accuracy. In this work we aim to review various machine learning methods used for this purpose, along with datasets and URL features used to train the machine learning models. The performance of different machine learning algorithms and the methods used to increase their accuracy measures are discussed and analyzed. The goal is to create a survey resource for researchers to learn the current developments in the field and contribute in making phishing detection models that yield more accurate results.

The year 2022 saw people's life being completely dependent on technology due to the global pandemic. Since digitalization became significant in this scenario, cyber criminals went on an internet crime spree. Recent reports and researches point to an increased number of security breaches that costs the victims a huge sum of money or disclosure of confidential data. Phishing is a cybercrime that employs both social engineering and technical subterfuge in order to steal personal identity data or financial account credentials of victims. In phishing, attackers counterfeit trusted websites and misdirect people to these websites, where they are tricked into sharing usernames, passwords,

banking or credit card details and other sensitive credentials. These phishing URLs may be sent to the consumers through email, instant message or text message. According to the FBI crime report 2022, phishing was the most common type of cyber-attack in 2020 and phishing incidents nearly doubled from 114,702 in 2019 to 241,342 in 2022. The Verizon 2022 Data Breach Investigation Report states that 22% of data breaches in 2022 involved phishing.

BACKGROUND:

1. Phishing Detection

A URL based phishing attack is carried out by sending malicious links, that seems legitimate to the users, and tricking them into clicking on it. In phishing detection, an incoming URL is identified as phishing or not by analysing the different features of the URL and is classified accordingly. Different machine learning algorithms are trained on various datasets of URL features to classify a given URL as phishing or legitimate.

2. Phishing Detection Approaches

In List Based approach, there are two lists, called whitelist and blacklist to classify legitimate and phishing URLs respectively. In, access to websites takes place only if the URL is in the whitelist. In blacklist is used. In Heuristic Based approach, the structure of a phishing URL is analysed. A pattern of URLs that were previously classified as phishing is created. URLs are classified according to their compliance with this pattern. The methods used to process the features of the URL plays a significant role in classifying websites accurately.

Visual similarity Based approach works by comparing the visual similarity of the website pages. Websites are classified as phishing or not by taking a server-side view of them as in. These two data are then compared with image processing techniques. Fake web pages are designed very close to

the original ones and it is easier to notice minor differences with image processing techniques, as users cannot notice them easily.

Content Based approach analyses the pages content. This method extracts features from page contents and third-party services like search engines and DNS servers. In authors proposed a detection method by specifying weights to the words that draw out from URLs and HTML contents. The words might include brand names that attackers use in the URL to make it look like a real one. Weights are specified according to their presence at different positions in URLs. The most probable words are chosen and then sent to Yahoo search to return the domain name with the highest frequency between the top 30 outcomes. The owners of the domain name are compared to decide if the website is phishing or not. In, they utilized a logo image to find the identity of web pages by matching real and fake web pages.

Fuzzy Rule based approach allows processing of ambiguous variables, then integrates human experts to classify those variables and relations between them. It is used to classify web pages based on the level of phishing that appears in the pages by employing a specific group of metrics and predefined rules. From the experimental results in the paper, for fuzzy logic systems, lower number of features leads to higher accuracy. If a fuzzy logic algorithm is affected by irrelevant features, the effectiveness of the classifier will decrease and vice-versa.

In Machine Learning based approach, machine learning models are created to classify a given URL as phishing or not using supervised learning algorithms. Different algorithms are trained on a dataset and then tested to learn the performance of each model. Any variations in the training data directly affects the performance of the model. This approach provides efficient techniques with high-performance for detecting phishing. This is a significant field of research and there are many papers that discuss machine learning based phishing detection.

3. Machine Learning Algorithms

There are several machine learning algorithms such as Naive Bayes, Decision Tree, Random Forest, Support Vector Machine, Logistic Regression and K-Nearest-Neighbour for detecting phishing websites. This is a very popular approach that has proved to be very efficient and accurate compared to other methods.

1.1 Purpose

Impacts of Phishing:

- Loss of Data
 - Clicking on a malicious link in an email can hand over the data and system of an organization to a hacker.
- Damaged Reputation
 - Companies suffer reputation loss following a data breach executed through phishing attacks.
- Direct Monetary Loss
 - Extra funds will be needed to manage identity protection, compensation of customers.
- Loss of Customers
 - Successful phishing attack scares customers away from a business.

Thus there is an immense need for the detection of phishing websites.

2. LITERATURE SURVEY

2.1 Existing problem :

As we have moved most of our financial, work related and other daily activities to the internet, we are exposed to greater risks in the form of cybercrimes.

2.2 References :

1.Title: Phishing Website Detection using Machine Learning Algorithms

- Authors: Rishikesh Mahajan MTECH Information Technology & Professor, Dept. Information Technology
- Publication : International Journal of Computer Applications (0975 – 8887) Volume 181 – No. 23, October 2018

2.Title: Detecting Phishing Websites Using Machine Learning

- Authors: Amani Alswailem computer Science Department, Al-Imam Muhammad Ibn Saud Islamic University, Riyadh, Saudi Arabia. Bashayr Alabdullah Computer Science Department, Al-Imam Muhammad Ibn Saud Islamic University, Riyadh, Saudi Arabia.
- Publication : 2019 2nd International Conference on Computer Applications & Information.

3. Title: Detection of Phishing Websites using Machine Learning

- Authors: Atharva Deshpande , Omkar Pedamkar , Nachiket Chaudhary ,
- Dr. Swapna Borde
- Publication : IJERT Volume 10, Issue 05 (May 2021)

4. Title: Phishing Website Detection Based on Deep Convolutional Neural Network and Random Forest Ensemble Learning

- Authors: Rundong Yang, Bin Wu
- Publication : Sensors 2021, 21, 8281

2.3 Problem Statement :

2.3.1 Discussion :

Problem - 1



Problem - 2



Problem - 3



Problem - 4



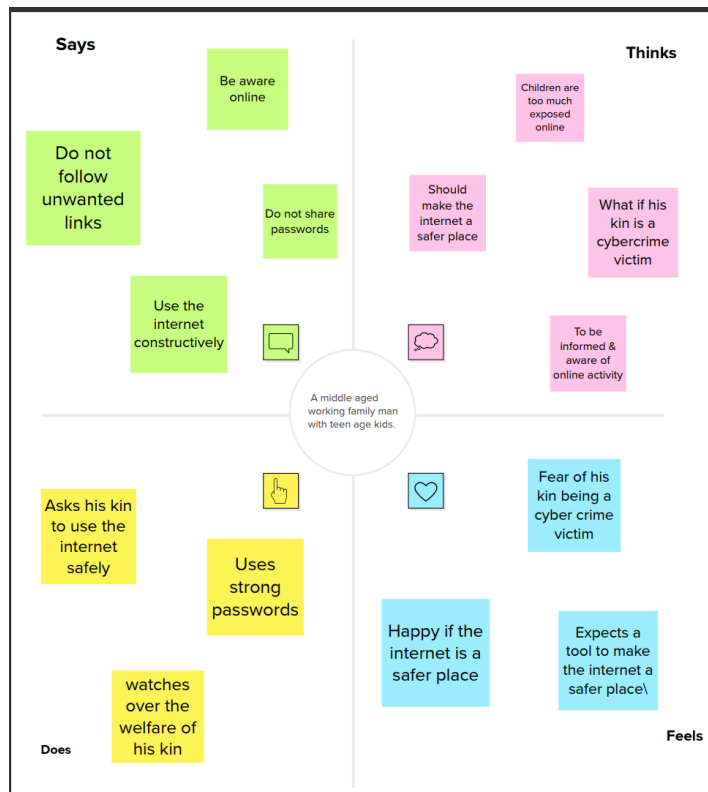
Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	Student	Opens an compromised link and enters the credentials	Gets Hacked	It was a Fake website	Insecure
PS-2	Shopkeeper	Opens a link from a bank and enters the account details	His bank account details and his credit card details gets hacked	The link was fake and the hacker aims for the money from the account.	untrustworthy
PS-3	Homemaker	make transactions for the purchased items from checkouts	Ended up as she purchases from an fake website	The website had an advertisement for discounts.	disappointed
PS-4	Employer	complete the tasks by the end of the day	Found an file through mail and opens it, and his company details gets hacked	the file seemed to be phishing and fake one	annoyed & disturbed

2.3.2 Definition:

To determine if an URL is “Legit” or a “phishing link” from an attacker.

3.IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas:



3.2 Ideation & Brainstorming :

Brainstorm

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

🕒 10 minutes

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

Shri Krishnaa

- Speaking of industries that have high online activity the OS & software used should be up to date.
- Highly sensitive information of the organization must be secured.
- Educate clients of the organization about the cyber crime.
- Educate people about doing safe on the internet (avoid this, starting with tag).

Siddharthan

- Speaking of attacks through e-mails, block the suspicious mail sender.
- Safe browsing should be practised.
- Focus on an extra line of security.
- Develop an antishishing technology.

Sanjeev

- Speaking of online transactions, should be more control on the who to choose.
- Using the "remember password" option should be avoided.
- Change transaction passwords periodically.
- Providing credit card details in unknown sites should be avoided.

Manivel

- Dataset of suspicious sites must be collected.
- The website names are to be studied to check its authenticity.
- Motive of the website is to be known.
- Basic knowledge on safe browsing on the internet is required.

Shankar Mahadevan

- Speaking of online shopping, secure platforms should be selected to purchase items/ goods.
- Secure internet connection is required.
- System alerts for suspicious activity alert should be developed.
- Employ an authenticity check.

Grouping ideas:



Prioritizing :

4

Prioritize

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

20 minutes



3.3 Proposed Solution:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Phishing has a list of negative effects on a business, including loss of money, loss of intellectual property, damage to reputation, and disruption of operational activities. These effects work together to cause loss of company value, sometimes with irreparable repercussions. Thus, we focus on creating a tool to detect the phishing websites
2.	Idea / Solution description	The idea is to detect/identify phishing websites thereby preventing/reducing the crimes done by such websites as a front
3.	Novelty / Uniqueness	The ML algorithm TF-IDF is in play to train the model for ensuring efficient prediction of phishing websites.
4.	Social Impact / Customer Satisfaction	By creating such a tool to detect the cloned websites, cybersecurity prevails in the online community. The tool to be created alerts the user if any suspicious activity is detected thus ensuring safety
5.	Business Model (Revenue Model)	Can be sold to third party email service providers
6.	Scalability of the Solution	Provides data integrity on the online community all over the globe .

3.4 Problem Solution fit:

Define CS, fit into CC	1. CUSTOMER SEGMENT(S) CS <ul style="list-style-type: none"> ● Business Organization ● Online Banking Sector ● Those who use Websites and URL 's for surfing through internet 	6. CUSTOMER CONSTRAINTS CC <p>Provides full access to scan the transaction process of the user and no breakdown of server connections</p>	5. AVAILABLE SOLUTIONS AS <p>This is applied to three different machine learning classifier - support vector machine, logistic regression and Naive Bayes. After training and testing the algorithms, it is observed that Naive Bayes classifier recorded the highest accuracy</p>	Explore AS, differentiate
	2. JOBS-TO-BE-DONE / PROBLEMS J&P <p>To identify the phishing sites and to protect users Credentials from hackers</p>	9. PROBLEM ROOT CAUSE RC <p>Having the data without any protection using anti phishing technologies, So that attacker creates fake website and steal the data.</p>	7. BEHAVIOUR BE <p>Customer finds the web phishing detection websites or applications and also the customer should provide all the transaction details of whole process .</p>	
Focus on J&P, use into BE, understand RC	3. TRIGGERS TR <p>Customer will get triggered because of data get stolen, theft of money and loss of privacy.</p>	10. YOUR SOLUTION SL <p>The links that gets checked for identifying phishing ,and we will be using various algorithm for making accurate prediction. Especially we are using Ada Boost Algorithm to make high accuracy prediction.</p>	8.CHANNELS of BEHAVIOR CH <p>8.1 ONLINE</p> <p>Pass the URL as input and identify whether it is a phishing site or not.</p> <p>8.2 OFFLINE</p> <p>Using the phishing detection application to predict the phishing sites in offline mode(offload the app).</p>	Identify strong TR & EM
	4. EMOTIONS: BEFORE / AFTER EM <p>BEFORE : Believing that the data is protected and secured in the Organization.</p> <p>AFTER : Feeling depressed as the data and money have been stolen.</p>			

4. REQUIREMENT ANALYSIS

4.1 Functional requirement:

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Registration (user's)	Manual registration (typing in the required details)
FR-2	Login	Successful Login with the registered e-mail Id and chosen password
FR-3	Creating a model	Model & train a ML algorithm to detect the phishing websites and compare them.
FR-4	Validation of the URL	URL is checked with the blacklisted set of URL's
FR-5	Integration	Flask is employed in the integration of front end & back end
FR-6	Ensure security	Notify the user through email or phone regarding the malicious website.

4.2 Non-Functional requirements:

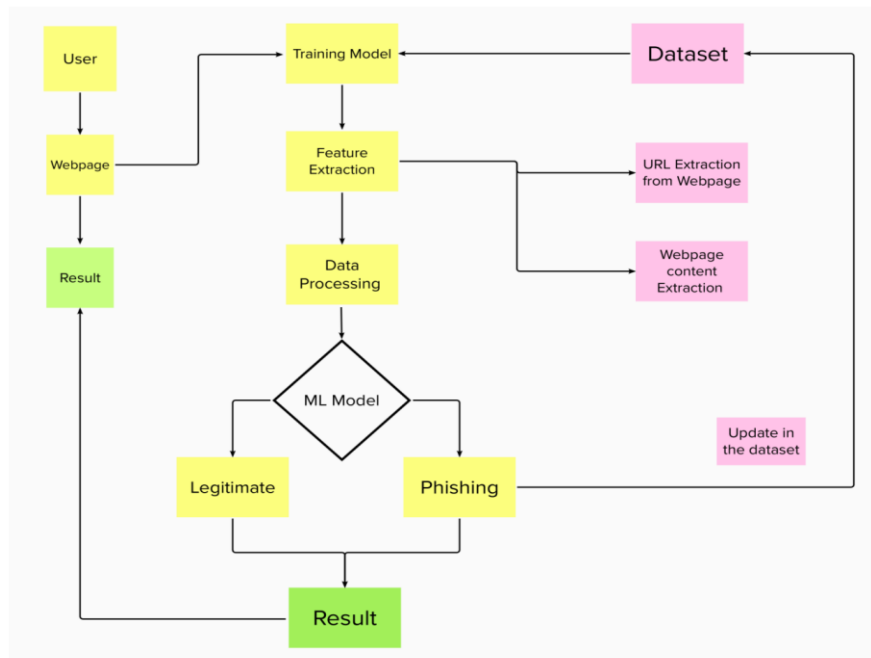
Non-functional Requirements:

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

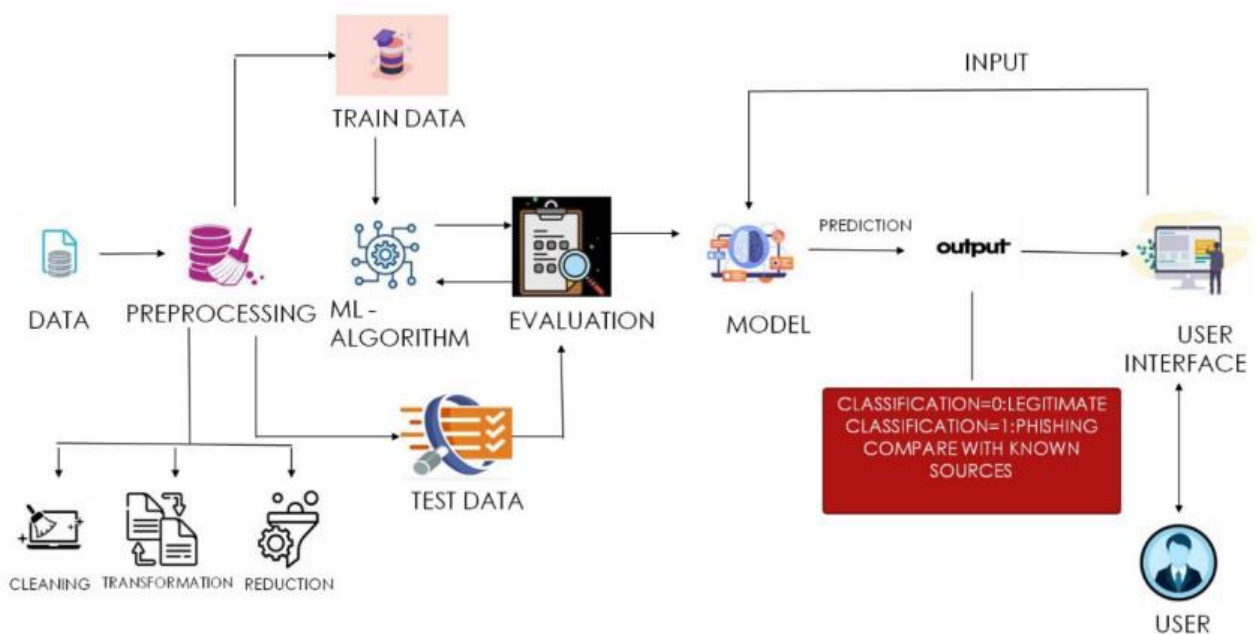
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	All URL's must be taken in by the tool for detection
NFR-2	Security	Security is ensured by alerting the user if any malicious link is detected
NFR-3	Reliability	The results provided by the tool must be of high efficiency
NFR-4	Performance	The tool created should be of high accuracy.
NFR-5	Availability	The tool must be readily available for the users to make use of it .
NFR-6	Scalability	Should be able to operate on a large scale

5. PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture



5.3 User Stories:

User Stories

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

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (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
	Login	USN-2	As a user, I can log into the application by entering email & password	Successful Login	High	Sprint-1
Customer (Web user)	User Input	USN-1	As a user, I can enter the required URL in the box while awaiting validation.	I can access the website without any problem	High	Sprint-1
Customer Care Executive	Feature Extraction	USN-1	In the event that nothing is discovered during comparison, we can extract features using a heuristic and a visual similarity technique.	As a user I can have comparison between websites for security	High	Sprint-1
Administrator	Prediction	USN-1	The model will use machine learning algorithms like a logistics regression and KNN to forecast the URLs of the websites.	I can accurately forecast the specific algorithms in this way.	High	Sprint-1
	Classifier	USN-2	To create the final product, I will now feed all of the model output to classifier.	I'll use this to identify the appropriate classifier for generating the outcome.	Medium	Sprint-2

6. PROJECT PLANNING & SCHEDULING

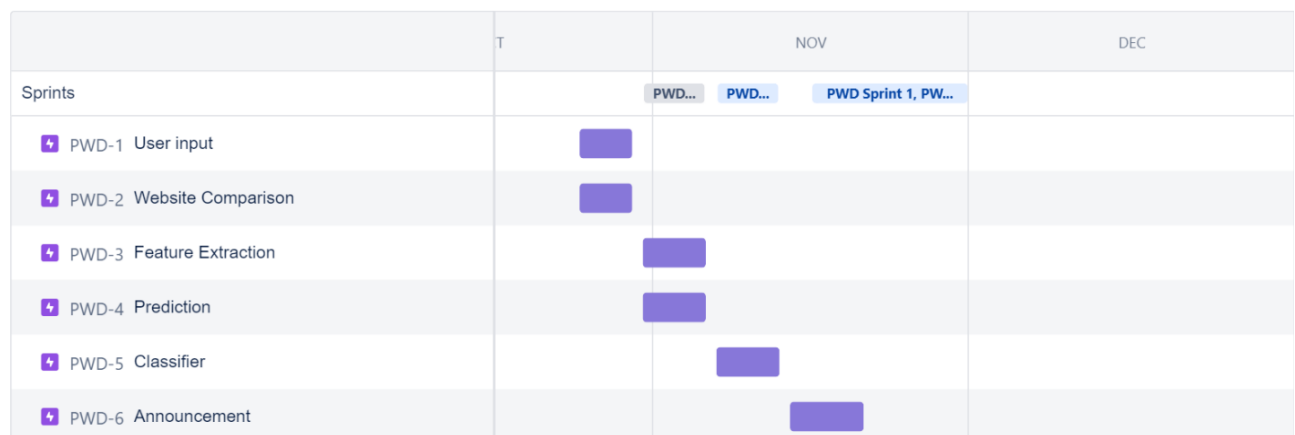
6.1 Sprint Planning & Estimation:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	User input	USN-1	User inputs an URL in the required field to check its validation.	1	Medium	Sanjeev K
Sprint-1	Website Comparison	USN-2	Model compares the websites using Blacklist approach.	1	High	Shri Krishnaa V N
Sprint-2	Feature Extraction	USN-3	After comparison, if none found on comparison then it extracts feature using similarity.	2	High	Manivel Prakash
Sprint-2	Prediction	USN-4	Model predicts the URL using Machine learning algorithms such as logistic Regression, KNN.	1	Medium	Shankar Mahadevan G
Sprint-3	Classifier	USN-5	Model sends all the output to the classifier and produces the final result.	1	Medium	Shankar Mahadevan G
Sprint-4	Announcement	USN-6	Model then displays whether the website is legal site or a phishing site.	1	High	Sanjeev K
Sprint-4	Events	USN-7	This model needs the capability of retrieving and displaying accurate result for a website.	1	High	Siddharthan S

6.2 Sprint Delivery Schedule

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

6.3 Reports from JIRA



Backlog



Epic

Insights

▼ PWD Sprint 3 7 Nov – 12 Nov (1 issue)

0 0 0 Complete sprint

PWD-12 Model sends all the output to the classifier and produces the final result.

IN PROGRESS

+ Create issue

▼ PWD Sprint 4 16 Nov – 19 Nov (2 issues)

0 0 0 Complete sprint

PWD-13 Model then displays whether the website is legal site or a phishing site.

IN PROGRESS

PWD-14 This model needs the capability of retrieving and displaying accurate result for a website.

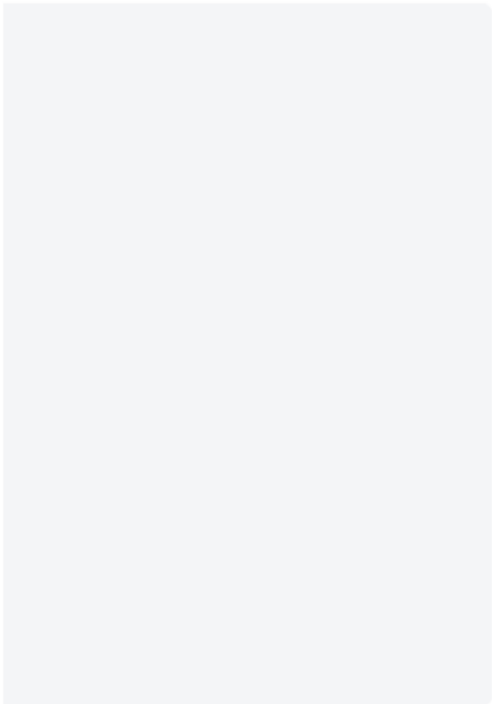
IN PROGRESS

+ Create issue

TO DO

IN PROGRESS 3 OF 3 ISSUES

▼ Issues without Epic 3 issues



Model sends all the output to the classifier and produces the final result.

PWD-12

Model then displays whether the website is legal site or a phishing site.

PWD-13

This model needs the capability of retrieving and displaying accurate result for a website.

PWD-14

7. CODING & SOLUTION

7.1 Classification of Website URLs:

```
import numpy as np
from sklearn.model_selection import KFold
kf = KFold(n_splits=5)
x, y = np.array(x), np.array(y)
kf.get_n_splits(x)

i = 0
for train_index, test_index in kf.split(x):
    X_train, X_test = x[train_index], x[test_index]
    y_train, y_test = y[train_index], y[test_index]
    xgb = XGBClassifier(learning_rate=0.4,max_depth=7)
    xgb.fit(X_train, y_train)
    i += 1
    print(f"Accuracy, fold {i} = ", accuracy_score(y_test, xgb.predict(X_test)))
```

```
Accuracy, fold 1 = 0.984622342831298
Accuracy, fold 2 = 0.9769335142469471
Accuracy, fold 3 = 0.9819086386250565
Accuracy, fold 4 = 0.9615558570782451
Accuracy, fold 5 = 0.9416553595658074
```

Web Phishing links come in different lengths and formats. Using a Natural Language Processing algorithm to process the website link alone, without taking into account any other data/meta-data about the website would thus be less performant.

So, we extract the following meta-data about the link each time a user enters a link, by scraping the data from Google:

- having_IP_Address
- URL_Length
- Shortening_Service
- having_At_Symbol
- double_slash_redirecting
- Prefix_Suffix
- having_Sub_Domain
- SSLfinal_State

- Domain_registration_length
- Favicon
- port
- HTTPS_token
- Request_URL
- URL_of_Anchor
- Links_in_tags
- SFH
- Submitting_to_email
- Abnormal_URL
- Redirect
- on_mouseover
- RightClick
- popUpWidnow
- Iframe
- age_of_domain
- DNSRecord
- web_traffic
- Page_Rank
- Google_Index
- Links_pointing_to_page
- Statistical_report

These 30 features are provided as input to a ML model – XGBoost classifier in our case. The model is trained on nearly 10K feature points and tested on 2K feature points.

7.2 ML Web Deployment

```
payload_scoring = {'input_data': [{'field': ['having_IPhaving_IP_Address', 'URLURL_Length', 'Shortining_Service', 'having_At_Symbol', 'double_slash_redirecting', 'Prefix_Suffix', 'having_Sub_Domain', 'SSLfinal_State', 'Domain_registration_length', 'Favicon', 'port', 'HTTPS_token', 'Request_URL', 'URL_of_Anchor', 'Links_in_tags', 'SFH', 'Submitting_to_email', 'Abnormal_URL', 'Redirect', 'on_mouseover', 'RightClick', 'popUpWidnow', 'Iframe', 'age_of_domain', 'DNSRecord', 'web_traffic', 'Page_Rank', 'Google_Index', 'Links_pointing_to_page', 'Statistical_report'], 'values': [values]}]}

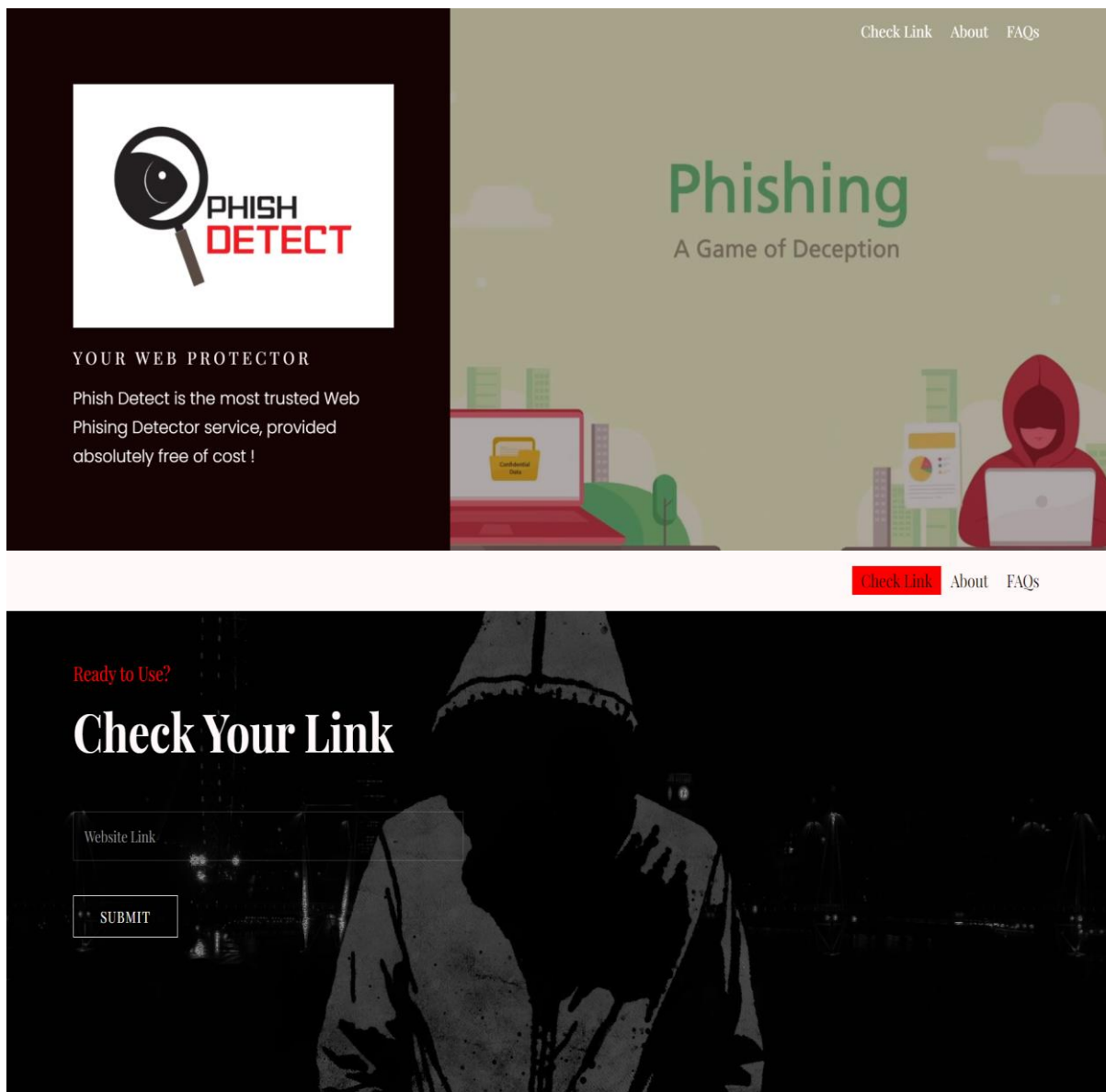
response_scoring = \
    requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/4a9dc193-0a81-4b5a-8675-a4c8291a818c/predictions?version=2022-11-19',
                  json=payload_scoring,
                  headers={'Authorization': 'Bearer ' + mltoken})
return response_scoring.json()['predictions'][0]['values'][0][0]
```

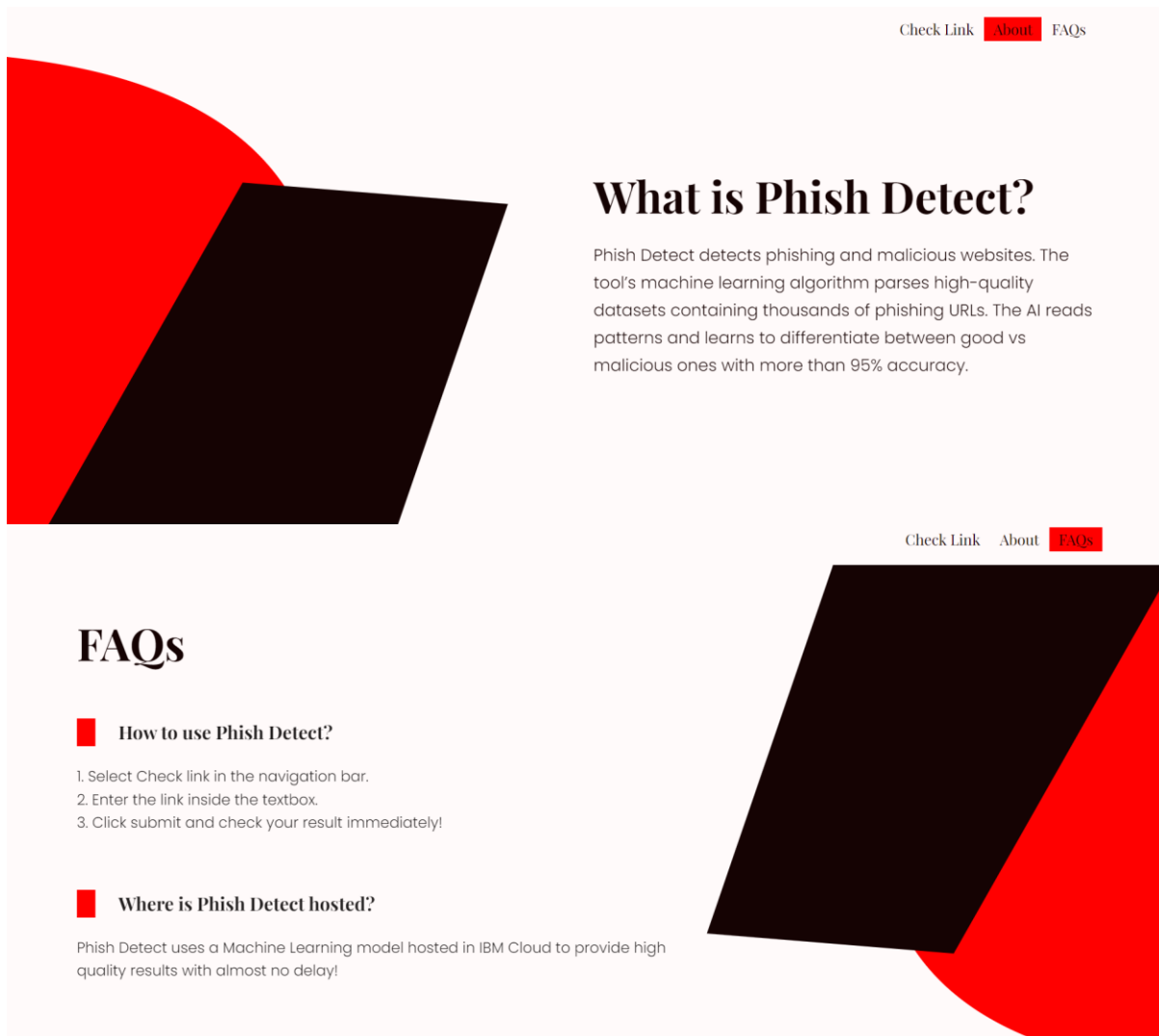
Phish Detect

Overview Assets Deployments Jobs Manage					
🔍 Search					
Name	Type	Status	Asset	Last modified	↓
(p) phish_detect_xgb	Online	🟢 Deployed	phish_detect	20 hours ago Shankar Mahadevan (You)	⋮

The XGBoost model is trained and deployed in IBM cloud. IBM cloud offers fast asynchronous processing of data, for multiple users simultaneously. This increases the scalability and efficiency of our solution.

7.3 Website – User Interface





The developed model is made available to the end user through our Phish Detect website. It is built using Bootstrap and Tailwind CSS, to provide an excellent mobile-friendly UI. It is hosted on a local development server built using Flask. When a link is entered by a user, the server sends a POST request to the hosted model, and tells whether the link is “Safe” or “Suspicious” based on the prediction.

8.TESTING

8.1 Test Cases

Locust Test Report

During: Invalid Date - Invalid Date

Target Host: http://127.0.0.1:5000

Script: PNT2022TMID21424.py

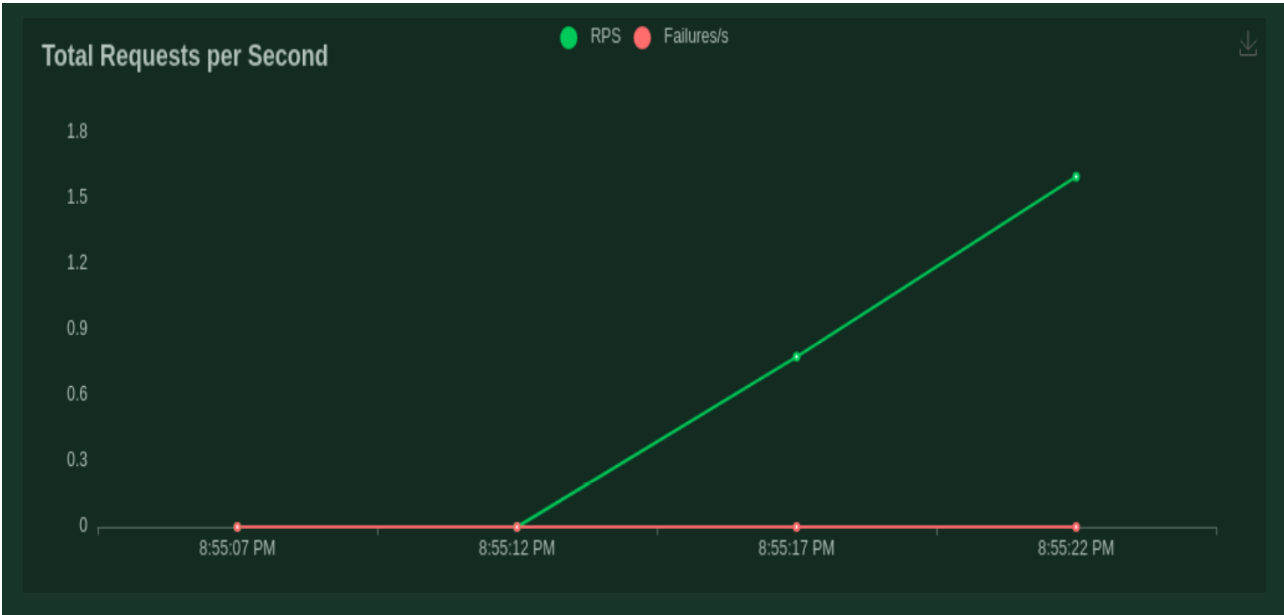
Request Statistics

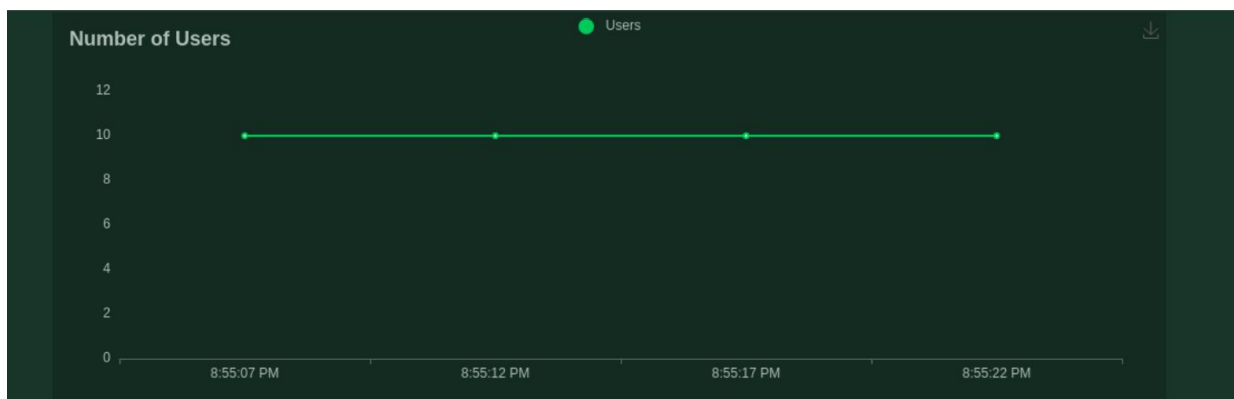
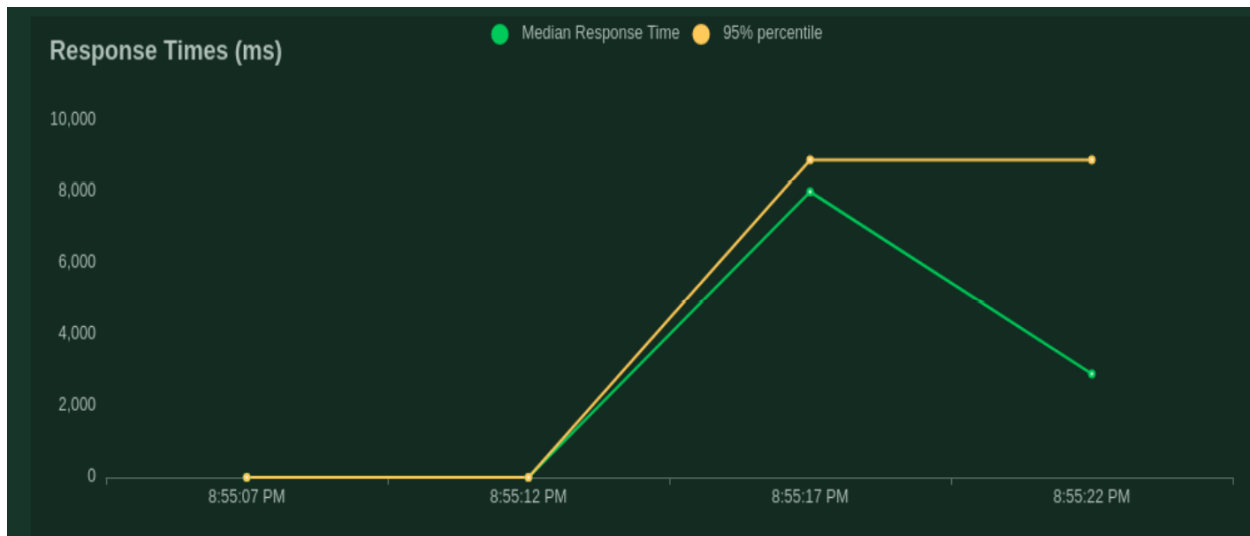
Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
POST	/y_predict	24	0	4648	2147	8893	22	1.2	0.0
	Aggregated	24	0	4648	2147	8893	22	1.2	0.0

Response Time Statistics

Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
POST	/y_predict	3300	4100	6900	8000	8200	8900	8900	8900
	Aggregated	3300	4100	6900	8000	8200	8900	8900	8900

Charts:





Final ratio

Ratio per User class

- 100.0% WebsiteTestUser
 - 100.0% poster

Total ratio

- 100.0% WebsiteTestUser
 - 100.0% poster

8.2 User Acceptance Testing

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	5	2	0	0	7
Duplicate	1	0	0	0	1
External	2	0	0	0	2
Fixed	8	2	0	0	8
Not Reproduced	0	0	0	0	0
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	8	2	0	0	10

Test Case Analysis

Section	Total Cases	Not Tested	Fail	Pass
Client Application	5	0	0	5
Security	2	0	0	2
Exception Reporting	1	0	0	1

Final Report Output	1	0	0	1
Version Control	1	0	0	1

9. RESULTS

9.1 Performance Metrics:

Decision Tree Model:

A decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements. Decision trees are commonly used in operations research, specifically in decision analysis, to help identify a strategy most likely to reach a goal, but are also a popular tool in machine learning.

Random Forest Model:

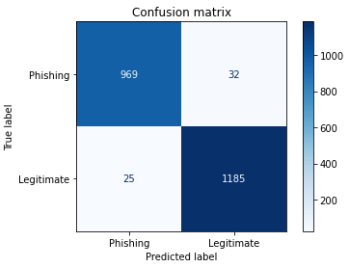
Random forests or random decision forests is an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned. Random decision forests correct for decision trees' habit of overfitting to their training set. Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees. However, data characteristics can affect their performance.

XG Boost:

XG Boost is an optimized distributed gradient boosting library designed to be highly **efficient**, **flexible** and **portable**. It implements machine learning algorithms under the Gradient Boosting framework. Boost provides a parallel tree boosting (also known as GBDT, GBM) that solve many data science problems in a fast and accurate way.

Model	Accuracy	Recall
Decision Tree	93%	92%
Random Forest	94%	93%
XGBoost	97%	97%

Model Performance Testing:

S.N o.	Parameter	Values	Screenshot
1.	Metrics	Classification Model: Confusion Matrix	

		<div>Accuracy Score</div> <div>Classification Report</div>	<div><pre>from sklearn.metrics import accuracy_score accuracy_score(y_test, y_test_xgb) 0.9742198100407056</pre></div> <div><table><thead><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr></thead><tbody><tr><td>0</td><td>0.97</td><td>0.97</td><td>0.97</td><td>1001</td></tr><tr><td>1</td><td>0.97</td><td>0.98</td><td>0.98</td><td>1210</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.97</td><td>2211</td></tr><tr><td>macro avg</td><td>0.97</td><td>0.97</td><td>0.97</td><td>2211</td></tr><tr><td>weighted avg</td><td>0.97</td><td>0.97</td><td>0.97</td><td>2211</td></tr></tbody></table></div>		precision	recall	f1-score	support	0	0.97	0.97	0.97	1001	1	0.97	0.98	0.98	1210	accuracy			0.97	2211	macro avg	0.97	0.97	0.97	2211	weighted avg	0.97	0.97	0.97	2211
	precision	recall	f1-score	support																													
0	0.97	0.97	0.97	1001																													
1	0.97	0.98	0.98	1210																													
accuracy			0.97	2211																													
macro avg	0.97	0.97	0.97	2211																													
weighted avg	0.97	0.97	0.97	2211																													
2.	Tune the Model	Hyperparameter Tuning - Validation Method – k-fold cross validation, where k=5	<div><pre>import numpy as np from sklearn.model_selection import KFold kf = KFold(n_splits=5) X, y = np.array(x), np.array(y) kf.get_n_splits(X) i = 0 for train_index, test_index in kf.split(X): X_train, X_test = X[train_index], X[test_index] y_train, y_test = y[train_index], y[test_index] xgb = XGBClassifier(learning_rate=0.05, max_depth=7) xgb.fit(X_train, y_train) i += 1 print(f"Accuracy, fold {i} = ", accuracy_score(y_test, xgb.predict(X_test))) Accuracy, fold 1 = 0.984622342831298 Accuracy, fold 2 = 0.9769335142469471 Accuracy, fold 3 = 0.9819086386250565 Accuracy, fold 4 = 0.9615558570782451 Accuracy, fold 5 = 0.9416553595658074</pre></div>																														

10. ADVANTAGES & DISADVANTAGES

Advantages:

1. Improve on Inefficiencies of SEG and Phishing Awareness Training

Secure email gateway's (SEG's) and phishing awareness training remain a critical tool in the fight against phishing and malware.

As increasingly-sophisticated phishing attacks, such as BEC, become more difficult to detect, even by trained security personnel. Thus there is an urgent need for the channel to provide customers with technology that not only strives to prevent intrusion, but can also help users after an attack has passed through the secure email gateway.

A mailbox-level anti-phishing solution offers an additional layer of protection by analyzing account information and understanding users' communication habits. This delivers an enhanced level of phishing protection to detect attacks faster, alert users and remediate threats as quickly as possible. Machine learning scores sender reputation enabling a baseline for what "normal communications" with a user should look like. It can then compare correspondence and incoming messages with multiple data points to identify and learn from anomalies.

2. It Takes a Load off the Security Team

Customers now have many tools on the market to enhance their email security. The best of these use artificial intelligence and machine learning to better identify some of the suspected threats. This not only improves security, but can significantly reduce the workloads of IT and security teams. According to a survey by Fidelis Cybersecurity, less than one in five organizations have a dedicated threat hunting team, and only half of those could handle more than eight investigations per day.

Security teams need all the help they can get, and must look beyond human intelligence to additional aids that protect the integrity of their corporate information. Automation can improve efficiency by flagging and analyzing the growing number of investigations, and by filtering out false positives. An automation detection and mitigation system reduces response time, identifies threats and can classify and remediate them with one click. It also delivers better metrics and intelligence with real-time insight into the strategies employed by the latest threats. Pushing these duties to an innovative solution not only increases security, but also enables security teams to focus more on policy and prevention.

3. It Offers a Solution, not a Tool

The goal of security isn't solely to bring tools to the table, it is to offer solutions. resellers ultimately need to ensure that solutions work for the organization, and that calls for listening to customers, understanding the channels, the issues and how they are impacting the organization.

Whereas basic tools simply offer information and basic applications, automated and advanced phishing threat protection solutions can help solve the challenges that customers face. This ultimately helps the channel discuss solutions with their customers and offer an overall picture and architecture of solving the challenges that are now inherent to email security. Automated advanced phishing threat protection defends against today's and tomorrow's threats with a system that continually learns and makes the entire organization more security conscious and aware.

4. Separate You from Your Competitors

As the channel is sometimes slow to move to some of the more advanced technologies, those that are fast will gain the upper hand and a

competitive advantage. Merely having these conversations will automatically distinguish you from many of your competitors. It demonstrates that you have a pulse on the latest threats and an insight into how artificial intelligence and machine learning can improve the security posture, without adding more burden to them.

This is especially important as Business Email Compromise attacks are on the rise. These attacks grew to record levels in 2017, according to the FBI, and can cost victimized organizations an average of \$25,000 to \$75,000. Trend Micro believes that business email compromise attacks will surpass \$9 billion in 2018. An analysis of 160 billion emails sent to 2,400 companies last year found that 89 percent of those companies were targeted with at least one attack.

Disadvantages:

- Real time updation of phishing links has not yet been implemented. It will help in making the model more robust.
- Deploying the model on a more powerful hardware cluster will make it more performant.

11.CONCLUSION

The objective of the project is to detect suspicious URL's that pretend to be genuine links ,which endure fatal loss to the innocent users. Cyber criminals have become experts at using sophisticated techniques to trick victims into sharing personal or financial information. But the best way to protect yourself is to learn how to spot a phishing scam before you take the bait. A URL based phishing attack is carried out by sending malicious links, that seems legitimate to the users, and tricking them into clicking on it. In phishing detection, an incoming URL is identified as phishing or not by analysing the different features of the URL and is classified accordingly. Different machine learning algorithms are trained on various datasets of URL features to classify a given URL as phishing or legitimate. Thus, phishing detection tool or any solution that prevents or mitigates the negative impacts of phishing on our society is much needed.

12. FUTURE SCOPE

In future if we get structured dataset of phishing, we can perform phishing detection much 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. Further enhancement of this work could be use of more advanced algorithms with 10-fold cross validation. Feature selection method can also be varied to see the effect on the varies parameters. A combination or hybrid machine learning algorithm can also be implemented to improve success rate and minimize false rate.

13. Source Code

13.1.1 Flask Server:

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
from flask import Flask, request, render_template
import InputScript
import requests

API_KEY = ''

def get_prediction_ibm(values):
    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': [
        'having_IPhaving_IP_Address',
        'URLURL_Length',
        'Shortining_Service',
        'having_At_Symbol',
        'double_slash_redirecting',
        'Prefix_Suffix',
        'having_Sub_Domain',
        'SSLfinal_State',
        'Domain_registration_length',
        'Favicon',
        'port',
        'HTTPS_token',
        'Request_URL',
        'URL_of_Anchor',
        'Links_in_tags',
        'SFH',
        'Submitting_to_email',
        'Abnormal_URL',
        'Redirect',
        'on_mouseover',
        'RightClick',
        'popUpWidnow',
        'Iframe',
        'age_of_domain',
        'DNSRecord',
        'web_traffic',
        'Page_Rank',
        'Google_Index',
        'Links_pointing_to_page',
        'Statistical_report',
    ]}], 'values': [values]}

    response_scoring = \
        requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/4a9dc193-0a81-4b5a-8675-
a4c8291a818c/predictions?version=2022-11-19',
                      json=payload_scoring,
                      headers={'Authorization': 'Bearer ' + mltoken})
    return response_scoring.json()[0]['predictions'][0]['values'][0][0]

app = Flask(__name__, template_folder='templates',
            static_folder='static')

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/predict')
def predict():
    return render_template('index.html')

success = ''
failure = ''

@app.route('/y_predict', methods=['POST'])
def y_predict():
    url = request.form['URL']
    checkprediction = InputScript.main(url)
    prediction = get_prediction_ibm(checkprediction[0])
    if prediction == 1:
        pred = 'This is a legitimate Website. You are Safe!'
        return render_template('index.html', success=pred)
    else:
        pred = 'This website looks suspicious. Be cautious!'
        return render_template('index.html', failure=pred)

if __name__ == '__main__':
    app.run(host='127.0.0.1', debug=True)
```

13.1.2 Link data Scraper:

<https://github.com/IBM-EPBL/IBM-Project-10243-1659122684/blob/main/Final%20Deliverables/Source%20Code/flask/inputScript.py>

13.1.3. ML Model Training:

https://dataplatform.cloud.ibm.com/analytics/notebooks/v2/a2f58e62-8396-4040-86a5-d6eb0b6d9615/view?access_token=7862b2a8fdce248d15cc1d94e44e1197c291ae71b54820a4133216f523bec3f5

13.1.4 Website Source code:

<https://github.com/IBM-EPBL/IBM-Project-10243-1659122684/blob/main/Final%20Deliverables/Source%20Code/flask/templates/index.html>

13.2 Project Deliverables - Github Link

GitHub Repo: <https://github.com/IBM-EPBL/IBM-Project-10243-1659122684>

Final Deliverables: <https://github.com/IBM-EPBL/IBM-Project-10243-1659122684/tree/main/Final%20Deliverables>

13.3 Video Demo:

<https://youtu.be/8Up7QHMXQbI>