WEB PHISHING DETECTION

PNT2022TMID27711

TEAM LEADER: VIGNESH S
TEAM MEMBER 1:VENKATESH R
TEAM MEMBER 2:SURYA A
TEAM MEMBER 3:SELVAM NARENDIRAN M

ABSTRACT:

Phishing is the fraudulent attempt to obtain sensitive information such as username, password, bank account details, and credit card details for malicious use. Phishing frauds might be the most popular cybercrime used today. There are various domain where phishing attacks can occurs like online payment sector, webmail and financial institution, file hosting or cloud storage and many others. The webmail and online sector was targeted by phishing more than in any other industry sector. Several anti-phishing techniques are there such because it is easy to use and implement but it fails to detect new phishing attacks machine learning is efficient techniques to detect phishing. It also removes drawback of existing approach. We perform detailed literature survey and proposed new approach to detect phishing website by features extraction and machine learning algorithm.

Key words : phishing detection, feature extraction ,phishing website ,phishing attacks

INTRODUCTION:

1. PROJECT OVERVIEW

There are a number of users who can purchase products in online and make payments through e-banking. There are e-banking websites that ask users to provide sensitive data such as username, password & credit card details, etc., often for malicious reasons. This type of e-banking website is known as a phishing website. Web service is one

of the key communications software services for the Internet. Web phishing is one of many security threat.

Common threats of web phishing:

- Web phishing aims to steal private information, such as usernames, passwords, and credit card details, by way of impersonating a legitimate entity.
- It will lead to information disclosure and property damage.
- Large organizations may get trapped in different kinds of scams.

This Guided Project mainly focuses on applying a machine-learning algorithm to detect Phishing websites.

- In order to detect and predict e-banking phishing websites, we proposed an intelligent, flexible and effective system that is based on using classification algorithms. We implemented classification algorithms and techniques to extract the phishing datasets criteria to classify their legitimacy. The e-banking phishing website can be detected based on some important characteristics like URL and domain identity, and security and encryption criteria in the final phishing detection rate. Once a user makes a transaction online when he makes payment through an e-banking website our system will use a data mining algorithm to detect whether the e-banking website is a phishing website or not.

2. PURPOSE

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

2. LITERATURE SURVEY

1. EXISTING PROBLEM

- Existing research works show that the performance of the phishing detection system is limited.
- Several anti-phishing techniques are there such because it is easy to use and implement but it fails to detect new phishing attacks.
- The problem of web phishing attacks has grown considerably in recent years and phishing is considered as one of the most dangers web to sophisticated techniques to attack and scam users all web has easily phishing by stealers.

3. PROBLEM STATEMENT DEFINITION

- Phishing is one of the techniques which are used by the intruders to get access to the user credentials or to gain access to the sensitive data .
- this type of accessing the is done by creating the replica of the websites which looks same as the original websites which we use on our daily basis but when a user click on the link he will see the website and think its original and try to provide his credentials.

- To overcome this problem we are using some of the machine learning algorithms in which it will help us to identify the phishing websites based on the feature present in the algorithm.

- By using these algorithm we can be able to keep the user personal credentials or the sensitive data safe from the intruders.

○ IDEATION AND PROPOSED SOLUTION

1. EMPATHY MAP CANVAS:

THINK AND FEEL:

- At present, visual similarities-based techniques are very useful for detecting phishing websites efficiently
- When the user, thinking the websites is a legitimate one, enter private information, and click the submit button, either a page cannot be found or not.

WHAT DO THEY HEAR :

- Phishing websites looks very similar in appearance to its corresponding legitimate websites.
- This methods uses the hyperlink function to check the legitimacy of web pages

WHAT DO THEY SEE :

- Phishing attacks often result in the theft of user data.
- Phishing attack gave become increasingly sophisticated and often transparently mirror the site being targeted.

WHAT THEY SAY AD DO :

- If the web page logs in successfully, it is classified as phishing other wise it undergoes further heuristic filtering.

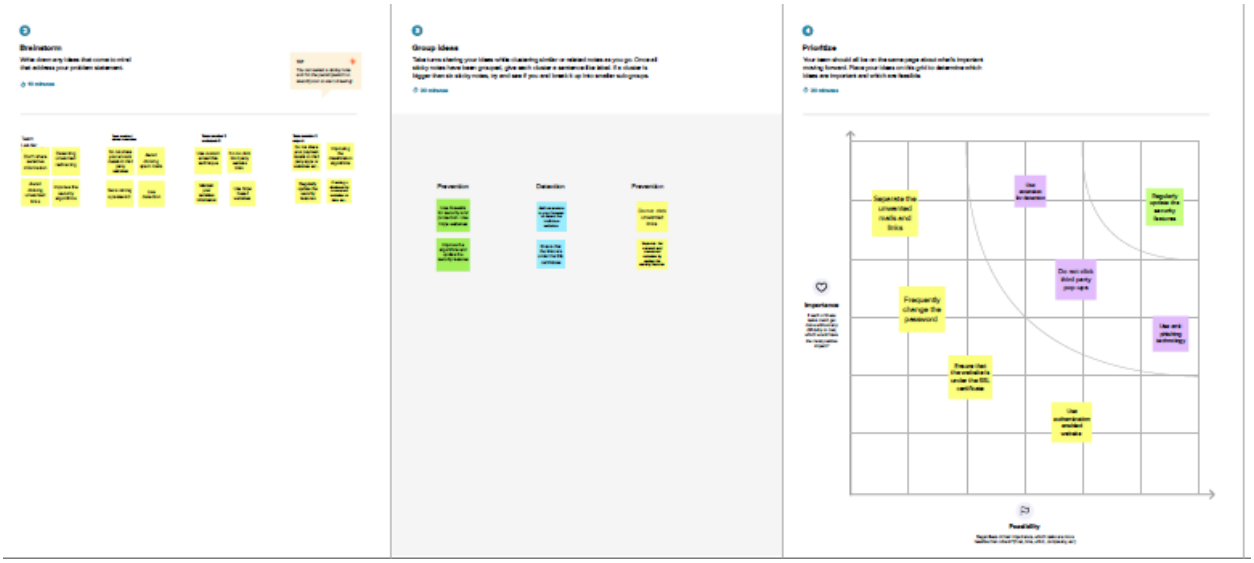
PAIN :

- Phishers steal personal information and financial account data.
- Detection of phishing attack with high accuracy has always been a challenging issue.

GAIN :

- With the significant growth of internet usage, people increasingly share their personal information.

IDEATION AND BRAINSTROMING :



PURPOSE :

- To find innovative solution to problem
- To leverage creativity and motivate to higher plateau of thinking
- Create the opportunity for expression of uncultivated ideas
- To draw from the diversity of new skills.

3. PROPOSED SOLUTION :

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

REQUIREMENT ANALYSIS :

FUNCTIONAL REQUIREMENTS

A function of software system is defined in functional requirement and the behavior of the system is evaluated when presented with specific inputs or conditions which may include calculations, data manipulation and processing and other specific functionality.

- Our system should be able to load air quality data and preprocess data.
- It should be able to analyze the air quality data.
- It should be able to group data based on hidden patterns.
- It should be able to assign a label based on its data groups.
- It should be able to split data into trainset and test set.
- It should be able to train model using trainset.
- It must validate trained model using test set.
- It should be able to display the trained model accuracy.
- It should be able to accurately predict the air quality on unseen data.

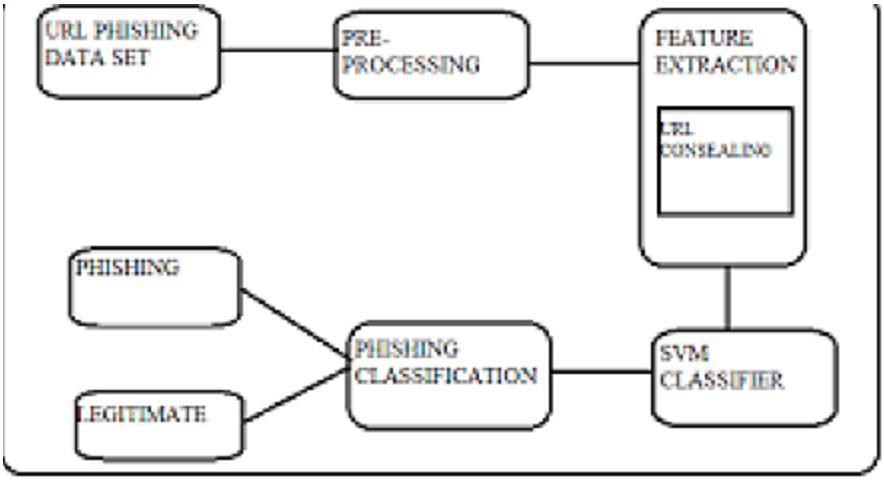
NON-FUNCTIONL REQUIREMENT

- Non-functional requirements describe how a system must behave and establish constraints of its functionality. This type of requirements is also known as the system's *quality attributes*. Attributes such as performance, security, usability, compatibility are not the feature of the system, they are a required characteristic. They are "developing" properties that emerge from

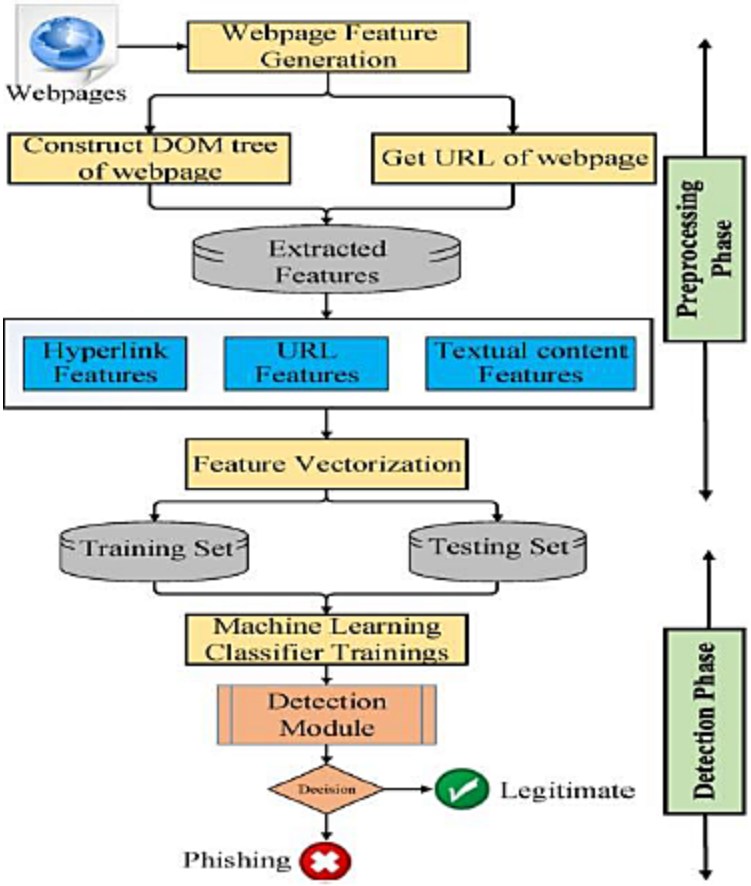
the whole arrangement and hence we can't compose a particular line of code to execute them.

5. PROJECT DESIGN :

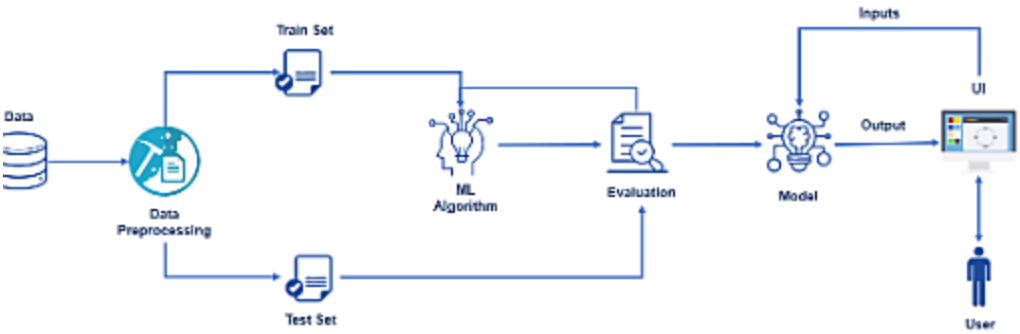
1. DATA FLOW DIAGRAM:



2. TECHNOLOGY STACK



SOLUTION ARCHITECTURE:



4. **USER STORIES :**

- As a user, I can enter the common threats of web phishing and I enter the username and password for detect phishing.

- As a user, I can predict the output by use of web frame flask.

- As a user, I can predict the phishing websites using of various techniques to develop ML models by console.

- The user can use the model by requesting the deployed model on cloud.

- As a user, I can so interested to using this websites with trained model on IBM.

- The user can Login / Sign in into the website by entering
  G-mail, Username and Create a password for future references.

1. **SPRINT PLANNING & ESTIMATION :**

Sprint	Requirements	Estimation	Team Members
Sprint-1	• Home page • URL Link	High	Vignesh S Venkatesh R
Sprint-2	• Import Flask • Model building	Medium	Selvam narendiran Surya A

Sprint-3	• Evaluate the model • Deploy the model on the cloud using IBM Watson • Integrate the training model on Flask	High	Vignesh S Venkatesh R
Sprint-4	• Testing	High	Selvam narendiran Surya A

2. SPRINT DELIVERABLE SCHEDULE:

Sprint	Sprint start date	Sprint end date	Story point	Sprint release date
Sprint-1	06 Oct 2022	09 Nov 2022	20	09 Nov 2022
Sprint-2	09 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-3	12 Nov 2022	15 Nov 2022	20	15 Nov 2022
Sprint-4	15 Nov 2022	18 Nov 2022	20	18 Nov 2022

7. CODING AND SOLUTIONS:

1. **FEATURE 1:**

DATA PRE-PROCESSING :

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import
confusion_matrix,accuracy_score ds=
pd.read_csv('dataset_website.csv')
ds.head()
d
s
.i
n
f
o
(
)
d
s
.i
s
n
u
ll
(
).
a
n
y
(
```

)
d
s
.
d
e
s
c
r
i
b
e
(
)
d
s
.
d
t
y
p
e
s
x
=
d
s
.i
l
o
c
[

:,
:
-
1
].
v
a
l
u
e
s
y
=
d
s
.
i
l
o
c
[
:
,
-
1
]
.
v
a
l
u
e

s

x

y

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=
0.2,random_state=0) from sklearn.linear_model import
LogisticRegression
lr=Log
isticRe
gressio
n()
lr.fit(x
_train,
y_trai
n)
y_pred
1=lr.pr
edict(x
_test )
from sklearn.metrics
import accuracy_score
lr=accuracy_score(y_tes
t,y_pred1)
lr
import pickle
pickle.dump(lr,open('phishi
ng_website.pkl','wb')
```

```
<!DOCTYPE
html>
                <html lang="en">
```

```
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Web Phishing detection</title>
<style>
body{
    background-image: linear-gradient(to right top, #082c5d, #00568c, #0081a4, #00
#09d17f);
    background-position: center center;
    background-attachment: fixed;
    background-repeat: no-repeat;
    background-size: auto;
}
p{
color:rgb(66, 66, 192);
}
h2{
    color:rgb(2, 179, 255);
}
.box{
    margin-top: 50px;
    margin-left: auto;
    margin-right: auto;
    background-color: #fafffe;
    padding: 10px;
    width: 30vw;
}

.input input{
    width: 20vw;
}
.lab{
    background-color:#000000;
    color:#ffffff;
}

</style>
</head>
<body>
<br>
<br>
<br>
<center><h1 style="color:rgb(25, 103, 136)">Phishing Detection</h1>
```

```
                <div class="box">
                <form action="." method="post" name="passdata">
                <div class="input">
                <label>Enter URL: </label>
                <input type="url" name="url" placeholder="enter url"required>
                </div>
                <br>
                <br>
                <input type="submit" value="Submit" class="lab">
                </form>
                <p>{{url}}</p>
                <h2>{{pred}}</h2>
                </div></center>
                </body>
                </html>

import numpy as
np
                from flask import Flask,request,render_template,jsonify
                import pickle


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

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

                @app.route('/',methods=['POST'])
                def y_predict():
                    url=request.form['url']
                    checkprediction=model.predict(checkprediction)
                    print(prediction)
                    output=prediction
                    if(output==1):
                       pred='You are safe!!  This is a Legitimate Website.'
                    else:
                        pred='You are on the wrong site. Be cautious!'
                    return
                render_template('final.html',prediction_text='{}'.format(pred),url=url)

                @app.route('/',methods=['POST'])
```

```
def predict_api():
    data=request.get_json(force=True)
    prediction=model.y_predict([np.array(list(data.values()))])
    output=prediction[0]
    return jsonify(output)

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

ADVANTAGES AND DISADVANTAGES:

Advantages:

- Flexible to use
- Low false positives
- Effective detection known phishing URL
- Have list of trusted senders
- Highest accurancy
- Easy to block IP address
- Check change on MAC address
- Identifies lies and true statements

DISADVANTAGES:

- Less efficient
- Detect phishing from known sender
- Cannot detect new phishing attacks
- Depends on standard databases

CONCLUSION:

- It is a outstanding that a decent enemy of phishing apparatus ought to anticipate the phishing assaults in proper timescale. We accept that the accessibility of a decent enemy of phishing device at the proper time scale is additionally imperative to build the extent of anticipating phishing sites. This

apparatus ought to be improved continually through consistent retraining. The procedure of finding the ideal structure is very difficult, and much of the time, this structure is controlled by experimentation. Our model takes care of the issues via computerizing the way towards organizing a neural system. Accomplishment is that our model uses a versatile system in structuring the system, while customary demonstrating procedures depend on experimentation. The preparation techniques utilized in our model since we attempt to improve the system execution. Our model have a speculation capacity of our model.

FUTURE SCOPE:

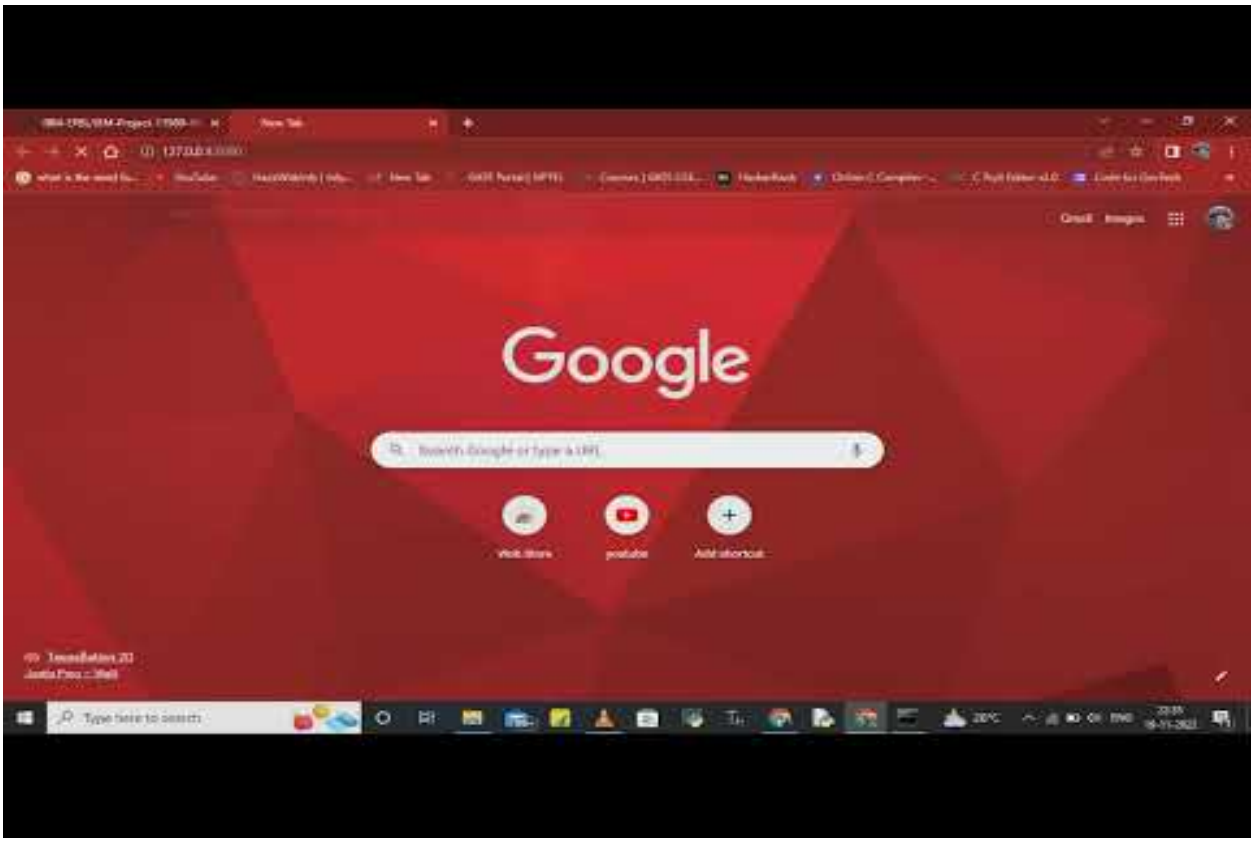
Further work can be done to enhance the model by using assembling models to get greater accuracy score. Ensemble methods is a ML techniques that combines many base models to generate optimal predict model. Further reaching future work would be combining multiple classifier, trained on different aspects of the same training set, into a single classifier that may provide a more robust prediction that any of the single classifiers on their own.

The methodology needs to be evaluated on how it might handle collection growth. The collections will ideally grow incrementally over time so there will need to be a way to apply a classifier incrementally to the new data, but also potentially have this classifier receive feedback the might modify it over time.

RESULTS:

GITHUB: https://github.com/

DEMO LINK:


Google
Gmail
Images
Web Store
youtube
Add shortcut
Type here to search