Detection-of-phishing-websites-from-urls-using-ibm-watson-studio

Group Members :-

Aditya Kunal Bhate - 19BCE10046
Shlok Sharma - 19BCE10102
Nidhi Gupta - 19BCE10423
Srushti Prashant Shintre - 19BCE100432

PROJECT REPORT

1. INTRODUCTION

   a. Overview

   Phishing is a common attack on credulous people by making them to disclosetheir unique information using counterfeit websites. The objective of phishing website URLs is to purloin the personal information like user name, passwords and online banking transactions. Phishersuse the websiteswhich are visuallyand semantically similarto those real websites. As technology continuesto grow, phishingtechniques started to progress rapidly and this needs to be prevented by using anti- phishing mechanisms to detect phishing. Machine learning is a powerful tool used to strive againstphishing attacks. This paper surveysthe features used for detectionand detection techniques using machine learning.
   We have developedour project using a websiteas a platform for all the users. This is an interactive and responsive websitethat will be used to detect whethera website is legitimate or phishing. This website is made using different web designing languages which include HTML, CSS and Python.

   b. Purpose

   Phishing is a common attack on credulous people by making them to disclosetheir unique information using counterfeit websites. The objective of phishing website URLs is to purloin the personal information like user name, passwords and online banking transactions. Phishersuse the websiteswhich are visuallyand semantically similarto those real websites. As technology continuesto grow, phishingtechniques started to progress rapidly and this needs to be prevented by

using anti- phishing mechanisms to detect phishing. Machine learning is a powerful tool used to strive againstphishing attacks. This paper surveysthe features used for detectionand detection techniques using machine learning.
Phishing is popularamong attackers, since it is easier to trick someoneinto clicking a malicious link which seems legitimate than trying to break through a computers defense systems. The malicious links within the body of the message are designedto make it appear that they go to the spoofed organization using that organizationslogos and other legitimate contents.

Here, we explain phishing domain (or Fraudulent Domain) characteristics, the featuresthat distinguish them from legitimate domains, why it is importantto detectthese domains, and how they can be detected using machine learning and natural language processing techniques.

There are a number of users who purchase products online and make payments through e-banking. There are e-bankingwebsites that ask users to providesensitive 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 oneof many security threats to web services on the Internet.
Common threatsof web phishing:

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

2. LITERATURE SURVEY

a. Existing Problem

Phishing is a cyber attack that uses disguised email as a weapon. The goal is to trickthe email recipient into believing that the message is something they want or needa request from their bank, for instance, or a note from someone in their company and to click a link or download an attachment.
What really distinguishes phishing is the form the message takes: the attackers masquerade as a trusted entity of some kind, often a real or plausibly real person, or a company the victim might do business with. It's one of the oldest types of cyberattacks, dating back to the 1990s, and it's still one of the most widespread

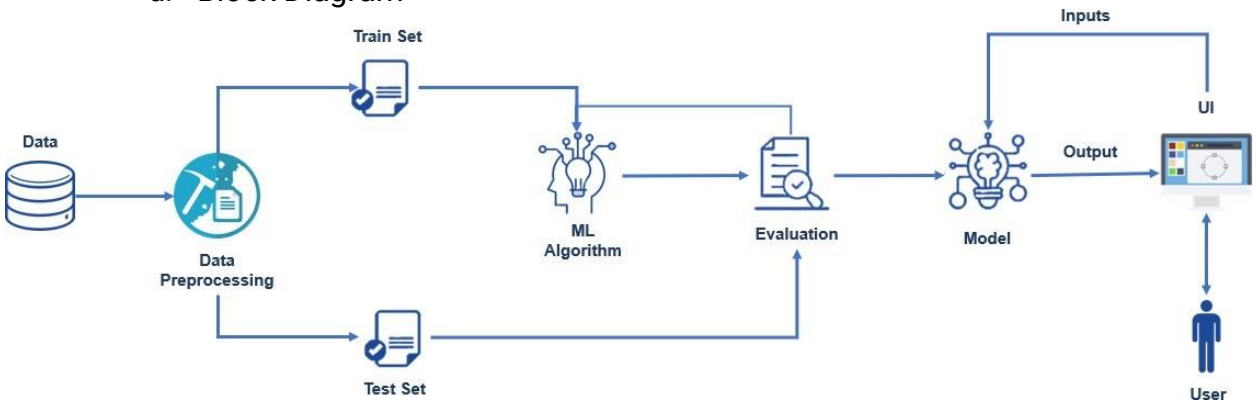
and pernicious, with phishing messagesand techniques becomingincreasingly sophisticated.

b. Proposed System

This Guided Project mainly focuses on applying a machine-learning algorithm to detectPhishing websites.In order to detectand 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 techniquesto extract the phishing datasets criteria to classify their legitimacy. The e-banking phishing website can be detectedbased on some important characteristics like URLand domain identity, and security and encryption criteria in the final phishing detection rate. Once a user makes a transaction online when he makes payment throughan e-banking websiteour system will use a data mining algorithm to detectwhetherthe e-banking website is a phishing website or not.

3. THEORATICAL ANALYSIS

a. Block Diagram



b. Hardware /Software Designing

i. Hardware

- 4GB RAM (minimum)
- 100GB HDD (minimum)
- Intel 1.66 GHz Processor Pentium 4 (minimum)
- Internet Connectivity

ii. Software

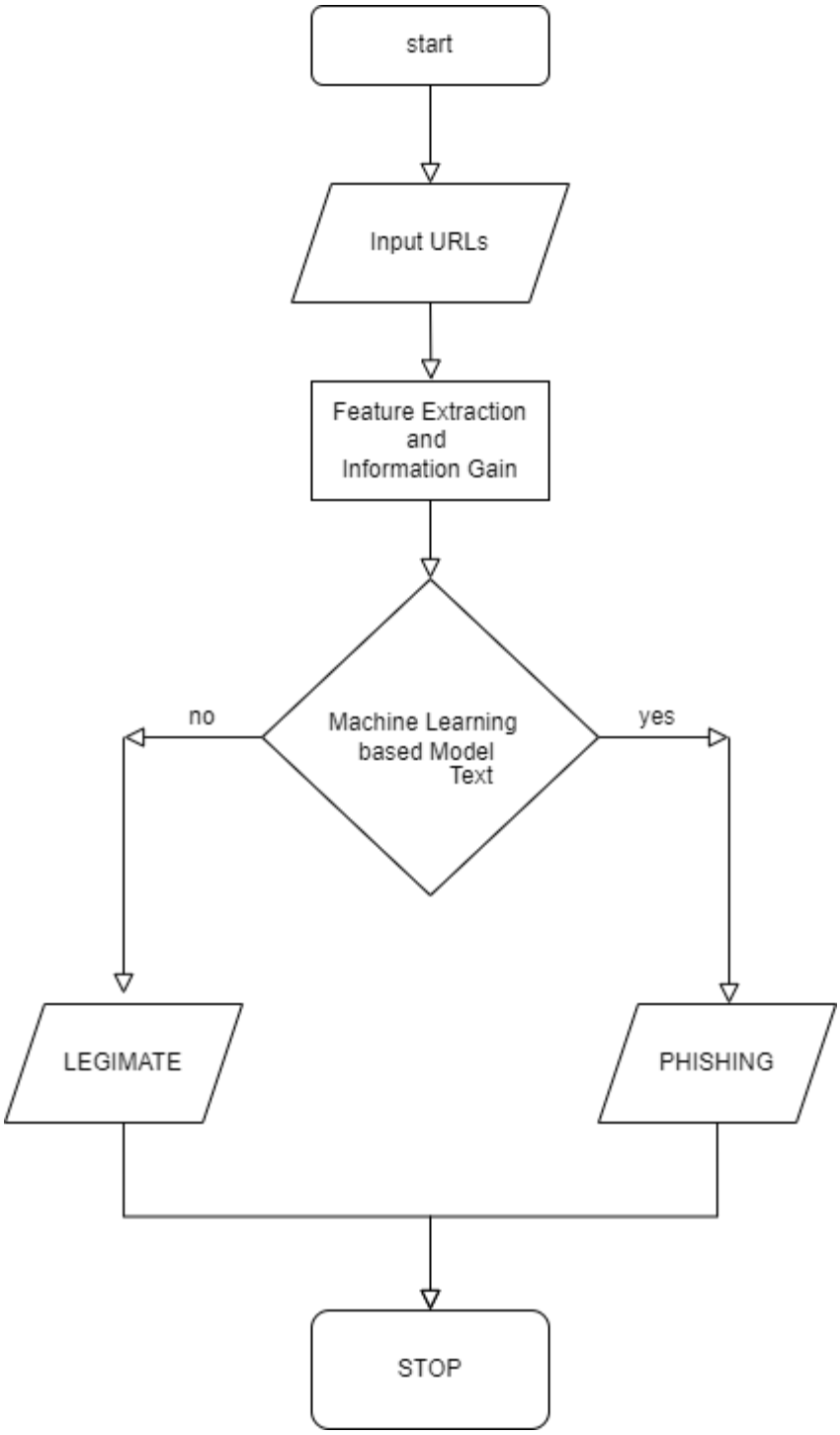
Anaconda Navigator: Anaconda Navigator is a free and open-source distribution of the Python and R programming languages for data science and machine learning-related applications. It can be installed on Windows, Linux, and macOS.Conda is an open-source, cross-platform, package management system. Anaconda comes with great tools like JupyterLab, Jupyter Notebook, QtConsole, Spyder, Glueviz, Orange, Rstudio, Visual Studio Code. For this project,we will be using Jupyter notebook and Spyder.

To build Machinelearning models you must require the following package

1. **Sklearn: Scikit-learn** is a libraryin Python that provides many unsupervised and supervised learning algorithms.
2. **NumPy: NumPy** is a Python packagethat stands for 'Numerical Python'.It is the core library for scientific computing, which contains a powerful n- dimensional array object
3. **Pandas: pandas** is a fast, powerful, flexible, and easy to use open-source data analysis and manipulation tool, built on top of the Python programming language.

4. **Matplotlib:** It providesan object-oriented API for embedding plots intoapplications using general-purpose,GUI,toolkits.
5. **Flask:** Web framework used for buildingWeb applications.

4. FLOWCHART



5. WORKING

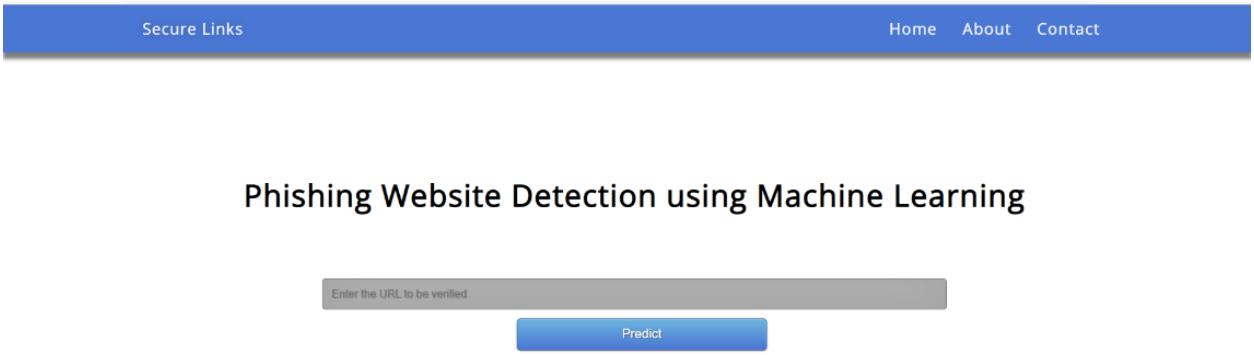
- We have collected data of URLs .
- In pre-processing, feature generation is done. These features are length of an URL, URL has HTTP, URL has suspicious character, prefix/suffix, number ofdots,

number of slashes, URL has phishing term, length of subdomain, URL contains IP address etc.

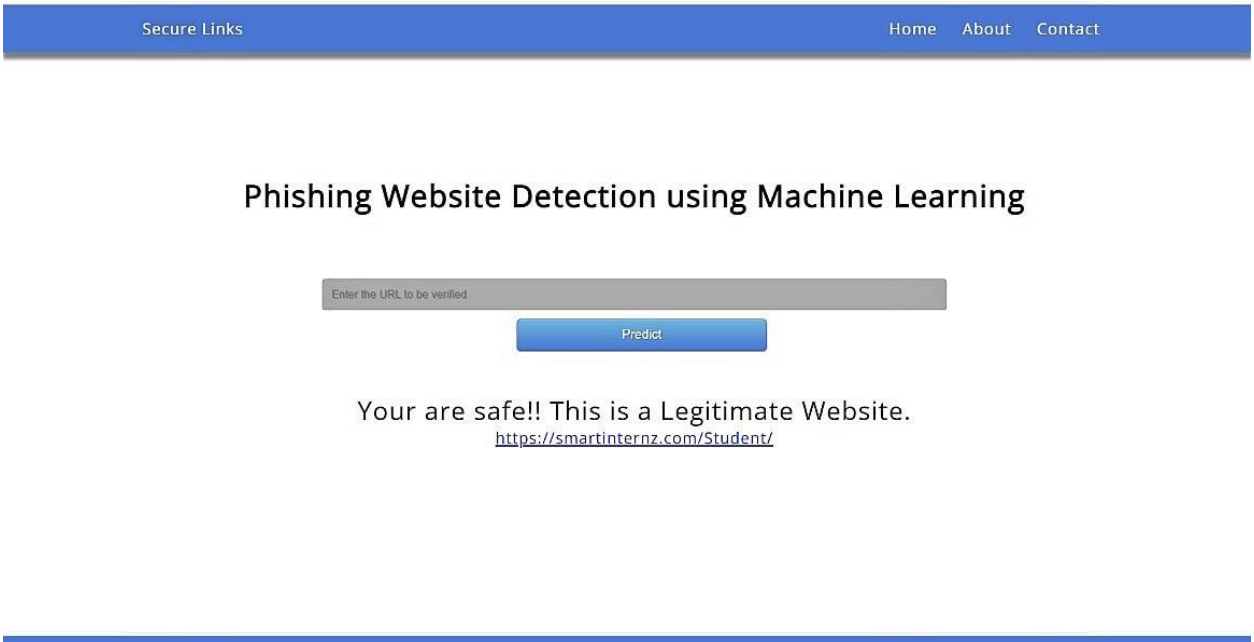
- After this, an organized dataset is made in which each detail incorporates the paired (0,1) which is then passed to the classifiers.
- Next, we train logistic regression classifier .
- Selected logistic regression as the best classifier which gives the most extremeprecision.
- Model deployment in cloud

6. HTML PAGES

   a. Main Page



   b. Outputs



5. App.py Code

```
import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle
#importing the inputScript file used to analyze the URL
import inputScript
import json
import requests
```

```
#load model
app = Flask(__name__)
model = pickle.load(open('Phishing_Website.pkl', 'rb'))

# @app.route('/')
# def helloworld():
#     return render_template("index.html")

# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = "Vhsx7vmc7em10KcOYiAVGpWhgtyR0-l5vzCXo8pT8EhI"
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": [{"fields": [array_of_input_fields], "values": [array_of_values_to_be_scored, another_array_of_values_to_be_scored]}]}

#response_scoring = requests.post('https://eu-gb.ml.cloud.ibm.com/ml/v4/deployments/8dd91582-83ea-48ee-b477-a85f1fbdf010/predictions?version=2022-03-06', json=payload_scoring, headers={'Authorization': 'Bearer ' + mltoken})
#print("Scoring response")
#print(response_scoring.json())
#Redirects to the page to give the user iput URL.
@app.route('/')
def predict():
    return render_template('final.html')

#Fetches the URL given by the URL and passes to inputScript
@app.route('/y_predict',methods=['POST'])
def y_predict():
    url = request.form['URL']
    checkprediction = inputScript.main(url)
    #t = [[checkprediction]]
payload_scoring = {"input_data": [{"fields":checkprediction , "values": checkprediction}]}
```

```
47.
48.     response_scoring = requests.post('https://eu-
    gb.ml.cloud.ibm.com/ml/v4/deployments/8dd91582-83ea-48ee-b477-
    a85f1fbdf010/predictions?version=2022-03-06', json=payload_scoring,
    headers={'Authorization': 'Bearer ' + mltoken})
49.     #print("Scoring response")
50.     #print(response_scoring.json())
51.     #prediction = model.predict(checkprediction)
52.
53.     #print(response_scoring)
54.     output=response_scoring.json()['predictions'][0]['values'][0][0]
55.     if(output==1):
56.         pred="Your are safe!!  This is a Legitimate Website."
57.
58.     else:
59.         pred="You are on the wrong site. Be cautious!"
60.     return render_template('final.html',
    prediction_text='{}'.format(pred),url=url)
61. #Takes the input parameters fetched from the URL by inputScript and
    returns the predictions
62. @app.route('/predict_api',methods=['POST'])
63. def predict_api():
64.     '''
65.     For direct API calls trought request
66.     '''
67.     data = request.get_json(force=True)
68.     prediction = model.y_predict([np.array(list(data.values()))])
69.
70.     output = prediction[0]
71.     return jsonify(output)
72.
73. if __name__== "__main__":
74.     app.run(debug=True)
75.
```

7. ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

- This system can be used by many E-commerce or other websitesin order to have goodcustomer relationship.

- Thissystem provides better performance as compared to other traditional classificationsalgorithms.
- Withthe help of this system user can also purchaseproducts online withoutany hesitation.

DISADVANTAGES:

- If Internet connection fails, this system won't work.
- All websites relateddata will be stored in one place.

8. APPLICATIONS

- Preventing loan application fraud
- Frauddetection in bankingand credit card payments
- Fraudprevention solutions in eCommerce
- Labelthe customers as fraud/not fraud

9. CONCLUSION AND FUTURE SCOPE

Thus to summarize, we have seen how phishingis a huge threat to the securityand safety of the web and how phishing detection is an important problem domain. We have reviewed some of the traditional approaches to phishing detection; namely blacklist andheuristic evaluation methods,and their drawbacks. We have testedone machine learningalgorithms on the Phishing Websites Dataset and reviewed results. We then built a flask application for detecting phishing web pages. The application allows users to detect whether a url id phishing website or not. We have detected phishing websites using Random Forest algorithm with and accuracyof 91%.

For future enhancements, we intend to build the phishing detection system as a scalableweb service which will incorporate online learning so that new phishing attack patterns

can easily be learned and improve the accuracy of our models with better feature extraction.

Although the use of URL lexical features alone has been shown to result in high accuracy (91%), phishers have learned how to make predicting a URL destination difficult by carefully manipulating the URL to evade detection. Therefore, combining these features with others, such as host, is the most effective approach.

10. BIBILIOGRAPHY

www.smartinternz.com

https://www.researchgate.net/publication/