PROJECT REPORT

Detection Of Phishing Websites From URLs

Submitted by

Amrithraj P H

1. INTRODUCTION

1.1 Overview

Phishing is a common attack on credulous people by making them to disclose their 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. 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. Machine learning is a powerful tool used to strive against phishing attacks. This paper surveys the features used for detection and detection techniques using machine learning.
We have developed our project using a website as a platform for all the users. This is 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 and Python.

1.2 Purpose

Phishing is a common attack on credulous people by making them to disclose their 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. 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. Machine learning is a powerful tool used to strive against phishing attacks. This paper surveys the features used for detection and detection techniques using machine learning.
Phishing is popular among attackers, since it is easier to trick someone into 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 designed to make it appear that they go to the spoofed organization using that organizations logos and other legitimate contents.

Here, we explain phishing domain (or Fraudulent Domain) characteristics, the features that distinguish them from legitimate domains, why it is important to detect these 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-banking websites that ask users to provide sensitive data such as username, password & credit card details, etc., often for

malicious reasons. This type of e-banking website is known as a phishing website. Web service is one of the key communications software services for the Internet. Web phishing is one of many security threats to web services on the Internet.

Common threats of web phishing:

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

2. LITERATURE SURVEY

2.1 Existing Problem

Phishing is a cyber attack that uses disguised email as a weapon. The goal is to trick the email recipient into believing that the message is something they want or need a request from their bank, for instance, or a note from someone in their company and to click a link or download an attachment.

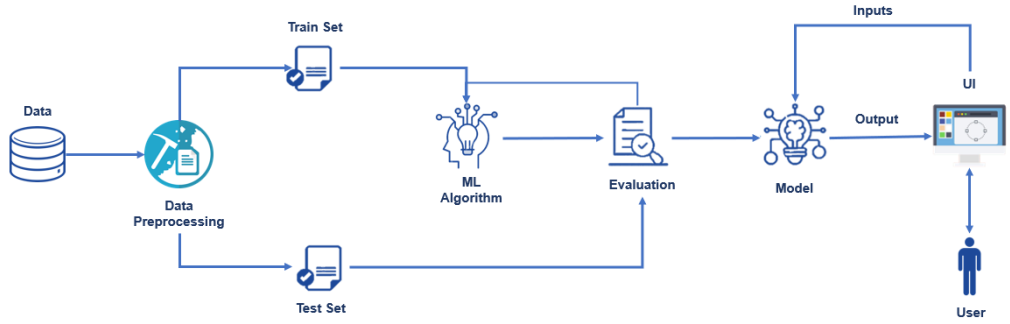
What really distinguishes phishing is the form the message takes: the attackers masquerade as a trusted entity of some kind, often a real or plausibly real person, or a company the victim might do business with. It's one of the oldest types of cyberattacks, dating back to the 1990s, and it's still one of the most widespread and pernicious, with phishing messages and techniques becoming increasingly sophisticated.

2.2 Proposed System

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.

3. THEORATICAL ANALYSIS

3.1 Block Diagram



3.2 Hardware /Software Designing

➢ Hardware

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

➢ Software

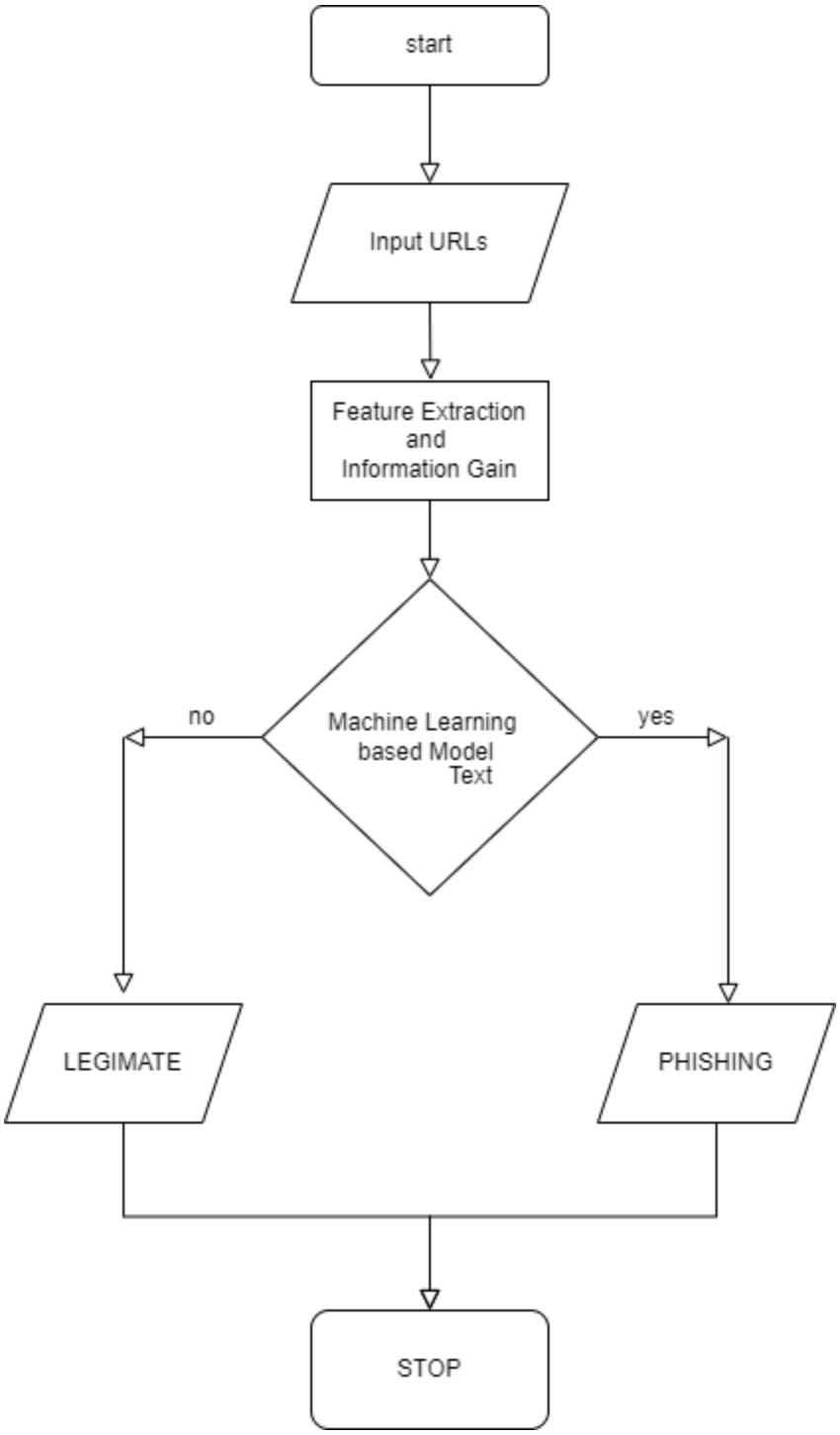
Anaconda Navigator: Anaconda Navigator is a free and open-source distribution of the Python and R programming languages for data science and machine learning-related applications. It can be installed on Windows, Linux, and macOS.Conda is an open-source, cross-platform, package management system. Anaconda 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 Machine learning models you must require the following package

- **Sklearn: Scikit-learn** is a library in Python that provides many unsupervised and supervised learning algorithms.
- **NumPy: NumPy** is a Python package that stands for 'Numerical Python'. It is the core library for scientific computing, which contains a powerful n-dimensional array object
- **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.

- **Matplotlib:** It provides an object-oriented API for embedding plots into applications using general-purpose,GUI,toolkits.
- **Flask:** Web framework used for building Web applications.

4. FLOWCHART

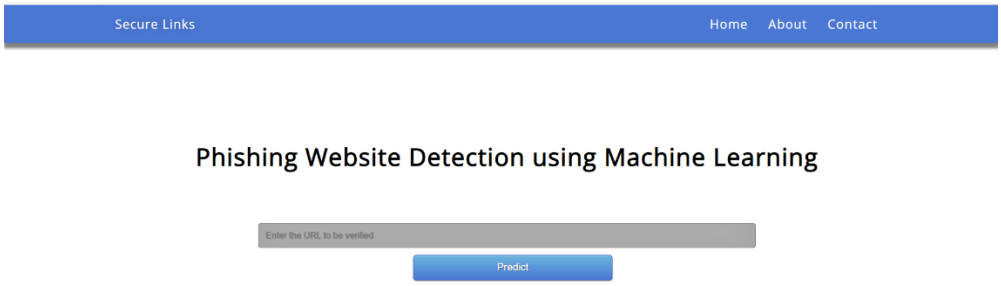

start
Input URLs
Feature Extraction
and
Information Gain
Machine Learning
based Model
Text
no
yes
LEGIMATE
PHISHING
STOP

5.WORKING

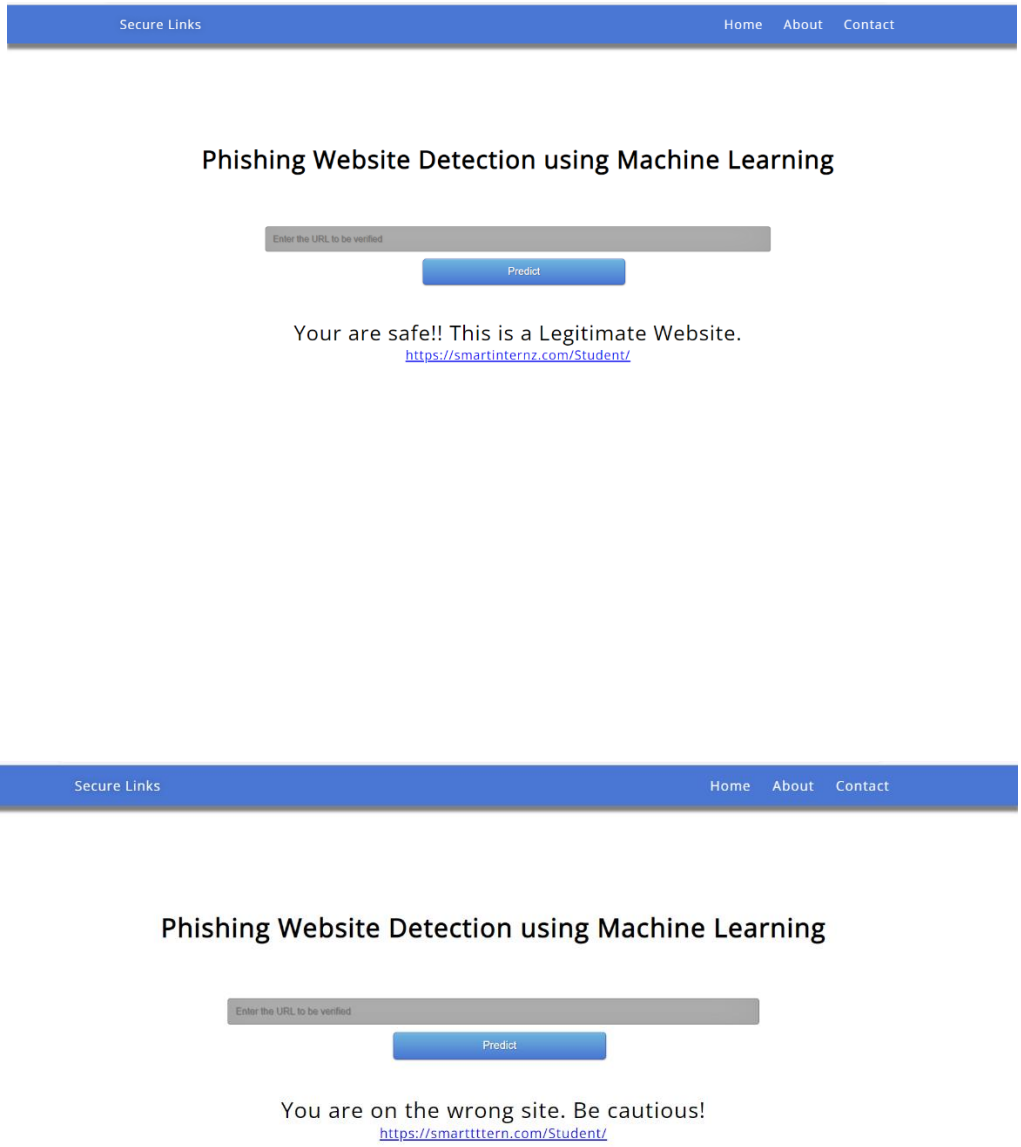
- We have collected data of URLs .
- In pre-processing, feature generation is done. These features are length of an URL, URL has HTTP, URL has suspicious character, prefix/suffix, number of dots, number of slashes, URL has phishing term, length of subdomain, URL contains IP address etc.
- After this, 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 extreme precision.
- Model deployment in cloud

6.HTML PAGES

6.1 Main Page



6.2 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 websites in order to have good customer relationship.

- This system provides better performance as compared to other traditional classifications algorithms.
- With the help of this system user can also purchase products online without any hesitation.

DISADVANTAGES:

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

8. APPLICATIONS

- Preventing loan application fraud
- Fraud detection in banking and credit card payments
- Fraud prevention solutions in eCommerce
- Label the customers as fraud/not fraud

9.CONCLUSION AND FUTURE SCOPE

Thus to summarize, we have seen how phishing is a huge threat to the security and 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 and heuristic evaluation methods, and their drawbacks. We have tested one machine learning algorithms 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 accuracy of 91%.

For future enhancements, we intend to build the phishing detection system as a scalable web 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/