Phishing Website Detection

Shaik Shahina[1], D. Murali[2]
[1]*PG Student, CSE, Quba College of Engineering & Technology*
[2]*Associate professor, CSE, Quba College of Engineering & Technology*

Abstract*— Online phishing is one of the most common attacks on the modern internet. The goal of phishing website uniform resource locators is to steal personal data including login credentials and credit card numbers. As technology keeps growing, phishing strategies began to develop rapidly. Machine learning built an effective device used to attempt phishing attacks. In this project, we have built a phishing website by using fastAPI. We have used two so many different libraries and two algorithms which are logistic regression and multimodal NP. The purpose of this project is to check whether phishing websites are good URLs or bad URLs. We gathered data to create a dataset of malicious links and curate it for the machine learning model.

Index Terms*— Phishing, Detection, API, URLs, Machine learning models, logistic regression.

I. INTRODUCTION

1.1 Introduction

In modern era Phishing becomes a main area of concern for security researchers due to the fact it is not tough to create the fake internet site which looks so close to legitimate internet site. Experts can discover fake web site7s however not all the customers can discover the fake website and such customers become the victim of phishing attack. Main purpose of the attacker is to steal banks account credentials. How hackers do their work, they send you just spam mail. In this mail though they will say that this email is mean to inform you that you're my university network password will expire in 24 hours and they have provide you to update the password and login when we click on that link we will redirect to that page which is a hacker server and they will be steal your data everything which is online. In our project we have to predict phishing websites whether they are good uniform resource locators (URLs) or bad URLs.

The set of phishing URLs are gather from open source service called Phish Tank. Benign URLs (uniform resource locator) with zero malicious detection were classified as benign and URLs with no less than eight detection were classified as malicious. It is being labeled as '0' and Phishing URL is being labeled as '1'. We study several machine learning algorithm for analysis of the characteristic in order to get a good understanding of the construction of the URLs that expand phishing. Phishing attacks are getting a success because lack of consumer awareness. Since phishing attack exploits the weaknesses found in customers, it's far very tough to mitigate them however it may be very vital to enhance phishing detection strategies. The general technique to discover phishing web sites through updating blacklisted URLs, Internet Protocol (IP) to the antivirus database which is also recognized as "blacklist" technique. To evade blacklists attackers makes use of innovative techniques to fool customers through modifying the URL to appear valid via obfuscation and lots of other easy techniques such as: fast-flux, in which proxies are automatically generated to host the web-page; algorithmic era of recent URLs etc. To entice human beings, Phisher sends "spooled" mails to several human beings as possible. When such emails are opened, the customers generally tend to redirect to the spoofed internet site.

The internet site intuitively asks you to run a software program or download a document while you're not waiting to do so. The internet site tells you that your device is inflamed with malware or that your browser extensions' or software program the machine of date. Malicious URLs on the website could be easily recognized by examining them through machine learning techniques.

1.2 Problem Statement

The major trouble is that phishing technique is bad accuracy and low adaptability to new phishing links. We plan to apply machine learning to overcome these limitation through imposing some classification algorithms and evaluating the overall performance of

these algorithms on our dataset. We have decided on the Random Forest method because of its excellent performance in classification but random forest and decision tree are not good with nlp data.

1.3 Objective

A phishing internet site is the most common social engineering approach that mimics trustful URLs and web pages. This project aims to predict phishing websites whether are good URLs or bad URLs. Both phishing and benign URLs of websites are collected to form a dataset and from them required URLs and these projects aimed functions are extracted. We have used so many different libraries and algorithms like Logistic Regression, numpy, pandas, MultinomialNB, Regexp Tokenizer many more. We gather data to create a dataset of malicious links and curate it for the ML model. The performance level of every model is measured and compared.

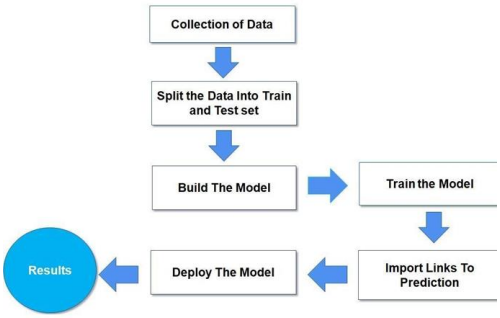

Figure 1.1 Flow of Method

II.LITERATURE SURVEY

Chunlin Liu, Bo Lang : Finding effective type for malicious URL detection : In ACM,2018 Chunlin et al. proposed method that primarily consciousness on individual frequency features. In this they've mixed statistical evaluation of URL with machine learning method to get end result this is more accurate for category of malicious URLs. Also they've compared six machine learning algorithms to confirm the effectiveness of proposed algorithm which offers 99.7% precision with fake positive rate much less than 0.4%.

Ankit Kumar Jain, B. B. Gupta : Towards detection of phishing websites on client-side using machine learning based approach :In Springer Science+Business Media, LLC, part of Springer Nature 2017

III. SYSTEM DEVELOPMENT

3.1 Analysis of the Algorithms

As our whole project is based on supervised machine learning. Supervised learning is a subcategory of machine learning and artificial intelligence. It works as we pass a data along with label to that data to a model and once the model is trained it recognizes some patterns and associates the labels to that pattern and thus makes the new predictions. There can be so many different applications possible using supervised learning. Some of them can be to detect the spam or spam detection. This is the way in which we can detect if mail is a spam or not if mail is a spam it will automatically put it in a spam folder and if it is not a spam mail then put it in your inbox. Another can be object classification and many more. Supervised machine learning has two categories:-

Classification: It helps find things that we can search by keywords, but it actually helps you find our own invention that's very close to our own. An area that is grouped in subject areas called classes and subclasses. Used to classify characteristics of invention. The type predictive modelling is the challenges of approximating the mapping characteristic from enter variables to discrete output variables. Example: email spam detector. The major purpose of the Classification algorithm is to pick out the category of a given dataset, and those algorithms are especially used to expect the output for the specific data. The algorithm which implements the kind on a dataset is referred to as a classifier.

Regression: It is a supervised learning method which enables in finding the correlation among variables and allows us to be looking ahead to the non-stop output variable primarily based totally at the simplest or greater predictor variables. It is specifically used for prediction, forecasting, time collection modeling, and figuring out the causal-impact relationship among variables. In Regression, we plot a graph among the variables which satisfactory suits the given data points, the use of this plot, the machine learning version could make predictions about the data.

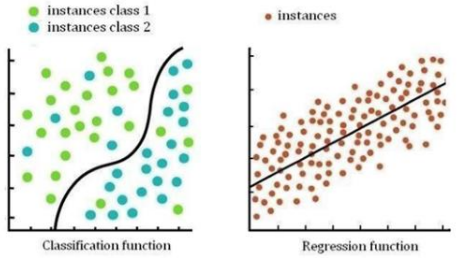

Fig 3.1 Classification and Regression

3.4 Algorithm

In this project, we have done various implementations for training and we have predicted phishing websites whether they are good URLs or bad URLs. As we have used some of the supervised algorithm. Firstly, we imported some libraries such as pandas, numpy, multinomialNB, LogisticRegression and many more. We have examined and pre-process the dataset and divide the datasets into training and test sets. Then we combine all the dataset into one frame.

After doing this with one rows we transform all the rows although all the URLs into tokenizer text. So it takes 4 seconds something so to convert all the rows more than 50 URLs into that word.

1. Print (‘getting words tokenizer’)

2. T0 is equal to time.perf underscore counter()

3.Phish underscore data [‘text underscore tokenizer’] is equal to phish underscore data.URL.map (Lambda t: tokenizer.tokenize(t))

4.T1 is equal to time.perf underscore counter()

5.Print (‘time taken’, t1,’sec’)

6. Phish underscore data.sample(5)

• So after converting into words we use a snowball. It is a NLTK API which is used to stream words.

• Stemmer is equal to SnowballStemmer (“English”)

• Print (‘getting words tokenizer’)

• T0 is equal to time.perf underscore counter()

• Phish underscore data [‘text underscore tokenizer’] is equal to phish underscore data [‘text_tokenized’].map (lambda 1: [stemmer. stem (word) for word in 1])

• T1 is equal to time.perf underscore counter() subtract T0 • Print (‘time taken’, T1,’sec’) All the holder that a words list are converted by stremmers. Then we just join the all the list words into just single sentence.

3.5 Model Development

In our project we have used FastAPI which is a python framework and import many libraries for different purposes. We have taken two algorithms which is LogisticRegression and MultinomialNB. LogisticRegression will predict the links are good or not and MultinomialNB work well with nlp data (natural language process). Then we have used some classification problems by using CountVectorizer and tokenizer. We have used someanother visualization. We can show that what is the hidden link in the phishing site which will redirect to another server. Then we have networkx it is creating a data structure, dynamic function and more. We are combining three datasets which we collected from several sites then we combine this dataset into one frame. The usability of this dataset is 10.0 which means very good. The data size is approx 30 mb.

IV. PERFOMANCE ANALYSIS

In this project, we have done various implementations for training and we have predicted phishing websites whether they are good URLs or bad URLs. As we have used some of the supervised algorithm.

Firstly, we imported some libraries such as pandas, numpy, multinomialNB, LogisticRegression and many more.

We have examined and pre-process the dataset and divide the datasets into training and test sets. Then we combine all the dataset into one frame.

```
In [2]: import pandas as pd # use for data manipulation and analysis
        import numpy as np # use for multi-dimensional array and matrix

        import seaborn as sns # use for high-level interface for drawing attractive and informative statistical graphics
        import matplotlib.pyplot as plt # It provides an object-oriented API for embedding plots into applications
        %matplotlib inline
        # It sets the backend of matplotlib to the 'inline' backend:
        import time # calculate time

        from sklearn.linear_model import LogisticRegression # algo use to predict good or bad
        from sklearn.naive_bayes import MultinomialNB # nlp algo use to predict good or bad

        from sklearn.model_selection import train_test_split # spliting the data between feature and target
        from sklearn.metrics import classification_report # gives whole report about metrics (e.g, recall,precision,f1_score,c_m)
        from sklearn.metrics import confusion_matrix # gives info about actual and predict
        from nltk.tokenize import RegexpTokenizer # regexp tokenizers use to split words from text
        from nltk.stem.snowball import SnowballStemmer # stemmes words
        from sklearn.feature_extraction.text import CountVectorizer # create sparse matrix of words using regexptokenizes
        from sklearn.pipeline import make_pipeline # use for combining all prerocessors techniques and algos

        from PIL import Image # getting images in notebook
        # from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator# creates words colud

        from bs4 import BeautifulSoup # use for scraping the data from website
        from selenium import webdriver # use for automation chrome
        import networkx as nx # for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks.

        import pickle# use to dump model

        import warnings # ignores pink warnings
        warnings.filterwarnings('ignore')
```

Fig 4.1 importing the libraries

```
In [2]: phishing_data1 = pd.read_csv('phishing_urls.csv',usecols=['domain','label'],encoding='latin1', error_bad_lines=False)
        phishing_data1.columns = ['URL','Label']
        phishing_data2 = pd.read_csv('phishing_data.csv')
        phishing_data2.columns = ['URL','Label']
        phishing_data3 = pd.read_csv('phishing_data2.csv')
        phishing_data3.columns = ['URL','Label']

In [3]: for i in range(len(phishing_data1.Label)):
            if phishing_data1.Label.loc[i] == '1.0':
                phishing_data1.Label.loc[i] = 'bad'
            else:
                phishing_data1.Label.loc[i] = 'good'
```

Fig 4.2 Combining dataset into one frame

After converting into words we used snowball it’s an nltk API (natural language toolkit) which is used to string words. It will remove all the English works and create some root words. Root words means that it will combine the common words like pictures, photos for this two words it will create the one word. Phishing data text streamer is equal to all the holder that list is words lists are converted into streamers. We also use word cloud. In our code we have used this to convert most repeated word into the word cloud form. Then we use chrome webdriver. This will create a new window of that chrome. So to this new chrome we will pass that link.

Fig 4.8 WordCloud

Fig 4.2.10 Common bad words

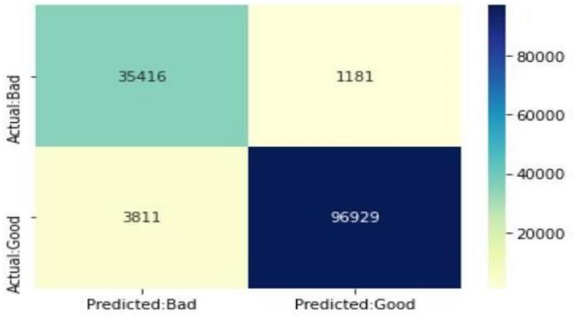

Fig 4.2.11 prediction

V. CONCLUSIONS

In our project we have used FastAPI which is a python framework and import many libraries for different purposes. We have taken two algorithms which is LogisticRegression and MultinomialNB. LogisticRegression will predict the links are good or not and MultinomialNB work well with nlp data (natural language process). Then we have used some classification problems by using CountVectorizer and tokenizer. We have used someanother visualization. We can show that what is the hidden link in the phishing site which will redirect to another server. Then we have networkx it is creating a data structure, dynamic function and more. We are combining three datasets which we collected from several sites then we combine this dataset into one frame. The usability of this dataset is 10.0 which means very good. The data size is approx 30 mb. The data contains more than 5 lakhs unique approach. The label column means that its prediction column in which there were two categories first is good and second is bad. After that we have checked the imbalanced of target column. Now we have a data, we convert URLs into vector form. We have used regular expression tokenizer which divide the string using regular expression. So, in our code we are just splitting only alphabets and some URLs have numbers, dots , slash etc which are not important our data. So we only gather the string and simultaneously we have transformed this in all the rows. After converting into words we used snowball it's an nltk API (natural language toolkit) which is used to string words. It will remove all the English works and create some root words. Root words means that it will combine the common words like pictures, photos for this two words it will create the one word. Phishing data text streamer is equal to all the holder that list is words lists are converted into streamers. Then we join all the lists words into single sentence. We also use word cloud. In our code we have used this to convert most repeated word into the word cloud form. Then we use chrome webdriver. This will create a new window of that chrome. So to this new chrome we will pass that link. Then by using beautifulSoup, we gather the all html code from its page source and it is getting all the anchor tags. So we will get the entire hidden link which will hacker use to redirect any users to this server and we create a data frame of this links. So it will give a two links : first is what we passed to this and second what are we getting from this link. Logistic regression object and we fit it by trainX, trainY. After that we checked the score and we are getting very good score which is 90.96. After that we just created the confusion matrix to see the actual prediction and normal prediction. Using Logistic Regression we are creating a pipeline. Then we are saving this pipeline model using pickle and we check the accuracy of it and it is giving very good accuracy.

VI. FUTURE ENHANCEMENTS

Through this project, one could recognize plenty approximately the phishing web sites and how they're differentiated from legitimate ones. This project may be taken in addition through developing browser extensions of growing a GUI. These have to classify the inputted URL to legitimate or phishing with the use of the stored model.

REFERENCES

[1] https://www.researchgate.net/profile/Rishikesh-Mahajan/publication/328541785_Phishing_Website_Detection_using_Machine_Learning_Algorithms/links/5d0397fd92851c9004394af4/Phishing-Website-

Detection-using Machine-Learning-Algorithms.pdf

[2] Ankit Kumar Jain, B. B. Gupta : Towards detection of phishing websites on client-side using machine learning based approach :In Springer Science+Business Media, LLC, part of Springer Nature 2017

[3] Chunlin Liu, Bo Lang : Finding effective classifier for malicious URL detection : In ACM,2018
https://www.researchgate.net/profile/Er- Purvi Pujara/publication/331198983_Phishing_Website_Detection_using_ Machine_Learning_A_Review/links/5c6bd4ae4585156b5706e727/P hishingWebsite-Detection-using Machine-Learning-A-Review.pdf

[4] Sahingoz, O.K., Buber, E., Demir, O. and Diri, B., 2019. Machine learning based phishing detection from URLs. Expert Systems with Applications, 117, pp.345-357.

[5] Sci-kit learn, SVM library. http://scikit-learn.org/stable/modules/svm.html.
https://www.unb.ca/cic/datasets/url-2016.html

[6] Ahmad Abunadi, Anazida Zainal ,OluwatobiAkanb: Extraction Process: A Phishing Detection Approach :In IEEE,2013