International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 05 | May - 2022 Impact Factor: 7.185 ISSN: 2582-3930

Machine Learning Technique to Detect Phishing Websites

Marisa Bhanu Venkata Sai
11809808
Computer Science and Engineering
Lovely Professional University
Jalandhar, India

Yarram Teja Krishna Reddy
11809847
Computer Science and Engineering
Lovely Professional University
Jalandhar, India

Kalla Samanth Chaithanya Akash
11812541
Computer Science and Engineering
Lovely Professional University
Jalandhar, India

Ravilla Mohan Sai
11809847
Computer Science and Engineering
Lovely Professional University
Jalandhar, India

Chikkala Sai Krishna
11812267
Computer Science and Engineering
Lovely Professional University
Jalandhar, India

Under the Guidance of Dr. Sudhanshu Prakash Tiwari

Abstract :

Phishing is one of the biggest cyber crime in World. Present now a days many of the people are getting attacked by these type of attacks. These attacks become very popular in these days. Attackers are sending emails, calls, text to someone who want to know the personal details, bank details, passwords and many more. Phishing is a type of social attack and it is often used to stole the user data , sensitive information , credentials , card numbers, etc....

This happens that when attacker send some malicious links to mobile or email in form of text like-

Example : you have been awarded 10 lakhs, to get them in to your bank account please click the following link. Then the victim get tricked by seeing all these links. Then victim get easily trapped into attackers bowl. Then victim click the link , then malicious software will be installed into victim phone, then the attacker can operate the victim phone and attacker can stole all sensitive information. It is one type of attack that victim cannot know that attacker stole all information. And when he clicks on the link then a bogus page which look like real and it can used to share its sensitive information without knowing to victim that this malicious page is specially designed for stealing information.

Introduction :

Phishing becomes most common cyber attack in world. It is a very serious cyber crime in whole world. And it become common. According to recent phishing statistics one in every 100 mails, is a phishing mail. And this leads to 4.8 mails per employee in their five day work week. The threat is very high in case of phishing attacks. The success rate of this attack make attackers more to do this. There is a phishing research publication named Avanan. According to Avanan statistics from 2016 to 2017 , there is increase of 65 to 70 percent in phishing attacks. And phishing attack become global problem in present days. And it affecting every region and economy in the global wise. While in 2018 many people in world wide nearly 85 percent of people received phishing attack that leads to more damage. This affects in many ways like productivity decreased by 67 percent, and there is loss of propriety data nearly by 54 percent. And it causes more damage to reputation nearly by 50 percent. This attack can be done in many ways like spear phishing , digital extortion, malware and credential harvesting. These all comes under phishing attacks. And it causes more damage by all these. Nearly 7.2 million dollars from spear phishing , 2.4 million dollars from malware, 5000 dollars from digital extortion and 400 dollars from credential harvesting per account. From the above statistic we can say phishing is one of the serious cyber crime in world. With these new phishing techniques machine learning is great technique to reduce these type of attacks to great extent. Using this we can load the model to the data set which contains features of phishing websites. By using this we can ask input in the form of URL and then we parse into the features of website and then we will test this data point and we will display output whether the given link is malicious or not.

Background Theory:

This section gives complete knowledge and understanding of different technologies and framework that were required in this project for its completion. The background Theory is listed below:

Python :

Python is one of the high level languages. And it is very easy to write. Compare to remaining programming languages syntax of python is very simple. It is called as Object oriented programming language with dynamic semantics. It is very useful language for application development. Why python is used for rapid application development means due to its high level in built data structures. And it is combination of dynamic binding and typing makes python more useful for application developing. It can also used as scripting language to connect the components that are existing together. Python syntax is very simple, easy to learn and it reduce the cost of maintenance of program. Python has vast number of modules and vast number of packages that makes program efficient and it encourages program modularity and code reuse. Python libraries are freely distributed and they are not chargeable. And Python interpreter is also available for free of cost. And they both can be available in the source or binary form.

Anaconda :

Anaconda is one of the major platform for distributing programming languages like python and R. And it is free of cost and open source platform for distribution. Anaconda provides major packages and vast number of libraries. And it also provide many number of packages related to machine learning and data science. One of the top most platform for distribution of the packages. And the main idea behind the anaconda platform is to make people very easy to use and install packages in single installation by anaconda.

Spyder :

Spyder is an integrated development environment called as IDE which is used for programming in python. And it is open source cross platform. It also integrated with vast number of built in packages in python stack. Most useful packages for scientific programming are NumPy, pandas, SymPy, IPython, Matplotlib and Cython. Spyder is a open source platform that released under MIT License. Spyder provides major support for interactive tools used for inspecting data. Spyder embeds python for quality assurance for specific code. It also provide introspection instruments like Pyflakes and Rope. Spyder is a huge cross platform that available through Windows, macOS and many linux distributions. It is available in macOS using Mac ports and available in big linux distributions through Arch linux, Debian, Gentoo linux, Ubuntu and many more. It is available through many platforms and it is special IDE used for programming in python.

Kaggle :

Kaggle is vast platform that supports variety of dataset publication formats. But in kaggle some data may not be accessible. So we are encouraging publishers to provide accessible and non proprietary format of dataset to upload in kaggle. It make customers to download datasets and very easy to access datasets. Not all accessible datasets are in platform and they are also easy for use in their works. It allows people to download largest datasets that can able to do with their projects and works. Datasets are not only simple but they are also repository. In each dataset there will be a place where you can discuss code and used to create your own projects. We can get very largest datasets of different sizes and shapes that these are very useful for projects all are available in Kaggle.

Objective :

The main objective of our below proposed project are listed below.

1. Reviewing the literature on several machine learning algorithms to find phishing websites related techniques.

2. Loading the data set and splitting the data set into train and test set.

3. Applying machine learning model and to train the data set on that.

4. To include the features of these websites after testing the data set and to save that model in a pickle file for further use.

5. Testing the developed tool and include necessary changes.

6. Filing a descriptive report by integrating the result.

Functional Requirements:

The functional Requirements for this project are mentioned below:

- FR1: The system should read the dataset.
- FR2: The system should import all the required libraries for the implementation.
- FR3: The system should extract features.
- FR4: Dataset is to be splitted into training set and testing set by system. Machine Learning Technique for detecting phishing websites.
- FR5: The system should build a model that fits the training dataset into the respective algorithm. • FR6: The system should predict the features of the phishing website using the model with the testing dataset.
- FR7: The system should give an accurate result.

Method and Methodology:

XG BOOST :

Before understanding of Xgboost , we have to discuss about decision tree. We discuss this later. XgBoost is a library which is created using c++ language used for optimising gradient boosting training. It is proposed as Extreme gradient boosting by researchers of Washington university.

Decision Tree :

A decision tree look like tree like structure which is eventually a flow chart where each internal node in a tree represent an attribute. And it has to be tested. And each branch in a tree represent the outcome of the attribute which is tested. Each terminal node also called as leaf node.. We split these sets based on an attribute value test. Recursive partitioning is a type of partitioning where this process is repeated on each subset which is derived in a recursive manner

Bagging :

Bagging is a type of classifier. It is used to form final prediction. It is a meta estimator used to fit base classifiers.. Meta estimator can be used in many ways. It is also used to reduce the variance of black box estimator. It reduces the variance by adding randomisation to its procedure and ensembles. Every base classifier is tested with training set simultaneously which is generated. It is

generated randomly by drawing or with replacing. The training sets of each base classifier is independent. From the real data many of them are recurred in outcoming training set and other may be left. Bagging is used to reduce variance which is also called as overfitting by voting or averaging.

Boosting :

Boosting is also a type of ensemble modelling. It is a technique to create a tough classifier from the poor classifiers. It can be done by building a big model in series by using a weak model. Firstly, model is created by training data and after by removing all errors in first model , then second model is built.

Gradient boosting :

It is also a boosting algorithm popularly using in these times. The technique that used in gradient boosting is that every predictor validates its predecessors mistakes. Instances trained are not tweaked , instead of that residual errors of predecessor as labels are used to train every predictor.

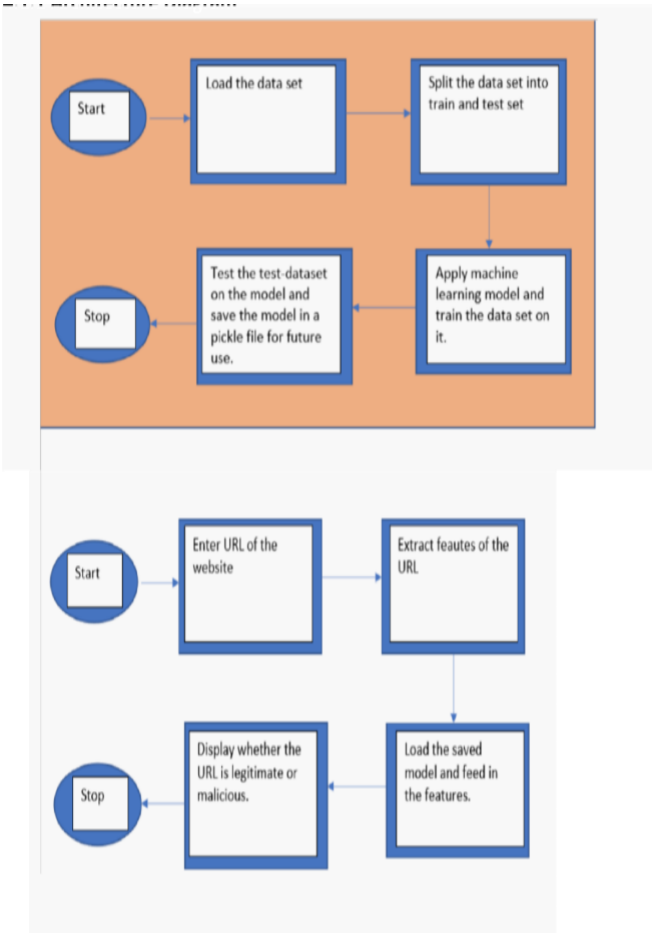
XG Boost :

We already discussed about gradient boosted decision trees. XGBoost is a result of gradient boosted decision trees. It plays main role in Kaggle. In this XGBoost, decision trees are made in a sequence which is also called as sequential form. In this algorithm weights has major role. Weights are given to all independent variables and make it in decision tree and predict correct outcomes. Weights which are predicted invalid by the first decision tree is raised and those are given to the second decision tree. Then all make a single classifiers. And these all single classifiers are combined which ultimately results in a strong model. This model is very precise. It can work on many prediction problems like classification.

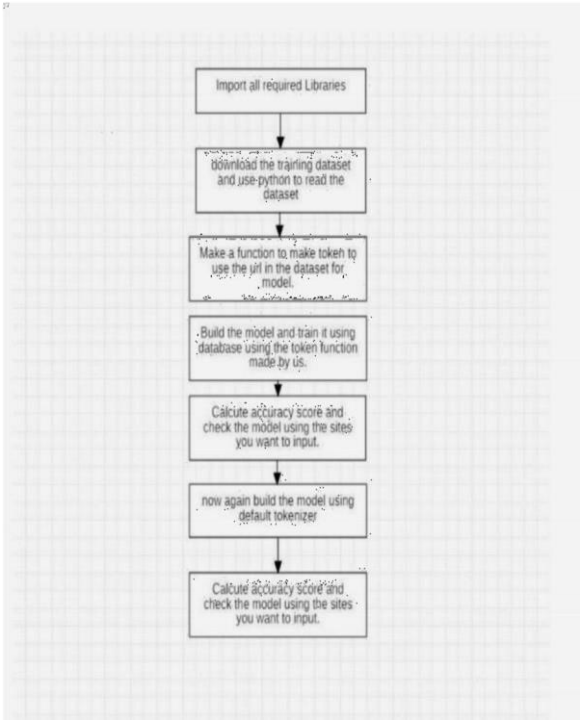
Design:

Design is helps when it comes to development since it is a form of blueprint for entire process from requirement making to end of project. Hence, in this section designs Machine Learning Technique for detecting phishing websites.

Architecture Diagram:



Flow Diagram:



Implementation:

```
#importing required packages for this module
import pandas as pd
```

```
#Loading the phishing URLs data to dataframe
data0 = pd.read_csv("C:\\Users\\sri divya\\Music\\DataFiles\\2.online-valid.csv")
data0.head()
```

2]:

	phish_id	url	phish_detail_url	submission_time	verified	verification_time	online
0	6557033	http://u1047531.cp.regruhosting.ru/acces-inges...	http://www.phishtank.com/phish_detail.php?phis...	2020-05-09T22:01:43+00:00	yes	2020-05-09T22:03:07+00:00	yes
1	6557032	http://hoysalacreations.com/wp-content/plugins...	http://www.phishtank.com/phish_detail.php?phis...	2020-05-09T22:01:37+00:00	yes	2020-05-09T22:03:07+00:00	yes
2	6557011	http://www.accsystemprblemhelp.site/checkpoint...	http://www.phishtank.com/phish_detail.php?phis...	2020-05-09T21:54:31+00:00	yes	2020-05-09T21:55:38+00:00	yes
3	6557010	http://www.accsystemprblemhelp.site/login_atte...	http://www.phishtank.com/phish_detail.php?phis...	2020-05-09T21:53:48+00:00	yes	2020-05-09T21:54:34+00:00	yes
4	6557009	https://firebasestorage.googleapis.com/v0/b/so...	http://www.phishtank.com/phish_detail.php?phis...	2020-05-09T21:49:27+00:00	yes	2020-05-09T21:51:24+00:00	yes

From the uploaded *Benign_list_big_final.csv* file, the URLs are loaded into a dataframe.

```
#Loading legitimate files
data1 = pd.read_csv("C:\\Users\\sri divya\\Music\\DataFiles\\1.Benign_list_big_final.csv")
data1.columns = ['URLs']
data1.head()
```

[6]:

	URLs
0	http://1337x.to/torrent/1110018/Blackhat-2015-...
1	http://1337x.to/torrent/1122940/Blackhat-2015-...
2	http://1337x.to/torrent/1124395/Fast-and-Furio...
3	http://1337x.to/torrent/1145504/Avengers-Age-o...
4	http://1337x.to/torrent/1160078/Avengers-age-o...

As stated above, 5000 legitimate URLs are randomaly picked from the above dataframe.

Final Data Sets:

In the above section we formed two dataframes of legitimate & phishing URL features. Now, we will combine them to a single dataframe and export the data to csv file for the Machine Learning training done in other notebook.

```
#Concatenating the dataframes into one
urldata = pd.concat([legitimate, phishing]).reset_index(drop=True)
urldata.head()
```

[38]:

	Domain	Have_IP	Have_At	URL_Length	URL_Depth	Redirection	https_Domain	TinyURL	Prefix/Suffix	DNS_Record	Web_Traffic	Domain_Age
0	graphicriver.net	0	0	1	1	0	0	0	0	0	1	0
1	ecnavi.jp	0	0	1	1	1	0	0	0	0	1	1
2	hubpages.com	0	0	1	1	0	0	0	0	0	1	0
3	extratorrent.cc	0	0	1	3	0	0	0	0	0	0	0
4	icicibank.com	0	0	1	3	0	0	0	0	0	1	0

```
urldata.tail()
```

[39]:

	Domain	Have_IP	Have_At	URL_Length	URL_Depth	Redirection	https_Domain	TinyURL	Prefix/Suffix	DNS_Record	Web_Traffic	Domai
25	decifra.com.br	0	0	1	3	0	0	0	0	0	1	
26	thejobnewsupdate.com	0	0	1	5	0	0	0	0	1	1	
27	mocktails.co.in	0	0	0	1	0	0	0	0	0	1	
28	chad-moore.com	0	0	0	1	0	0	0	1	0	1	
29	thejobnewsupdate.com	0	0	1	5	0	0	0	0	1	1	

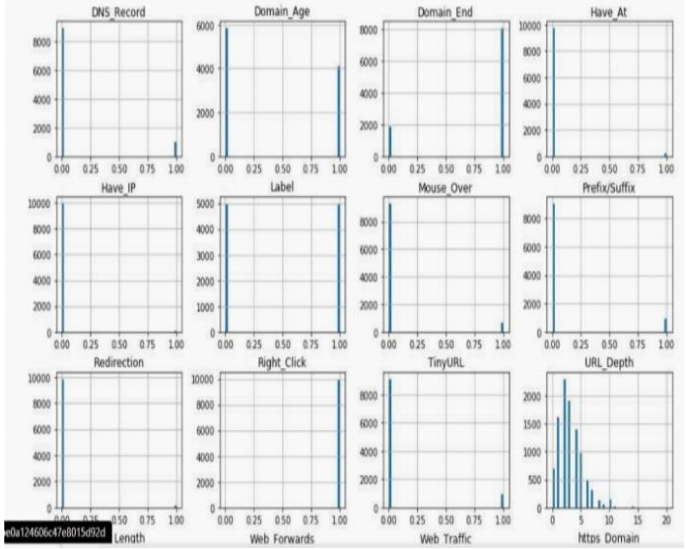
The reuted csv file is uploaded to this notebook and stored in the dataframe.

```
#importing basic packages
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

```
#Loading the data
data0 = pd.read_csv('C:\\Users\\sri divya\\Music\\DataFiles\\5.urldata.csv')
data0.head()
```

	Domain	Have_IP	Have_At	URL_Length	URL_Depth	Redirection	https_Domain	TinyURL	Prefix/Suffix	DNS_Record	Web_Traffic	Domain_Age
0	graphicriver.net	0	0	1	1	0	0	0	0	0	1	1
1	ecnavi.jp	0	0	1	1	1	0	0	0	0	1	1
2	hubpages.com	0	0	1	1	0	0	0	0	0	1	0
3	extratorrent.cc	0	0	1	3	0	0	0	0	0	1	0
4	icicibank.com	0	0	1	3	0	0	0	0	0	1	0

```
#Plotting the data distribution
data0.hist(bins = 50,figsize = (15,15))
plt.show()
```



Splitting Data:

```
# Seprating & assigning features and target columns to X & y
y = data['Label']
X = data.drop('Label',axis=1)
X.shape, y.shape
```

```
((10000, 16), (10000,))
```

```
# Splitting the dataset into train and test sets: 80-20 split
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 = 12)
X_train.shape, X_test.shape
```

```
((8000, 16), (2000, 16))
```

```
# Decision Tree model
from sklearn.tree import DecisionTreeClassifier

# instantiate the model
tree = DecisionTreeClassifier(max_depth = 5)
# fit the model
tree.fit(X_train, y_train)
```

```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
                       max_depth=5, max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort='deprecated',
                       random_state=None, splitter='best')
```

```
#predicting the target value from the model for the samples
y_test_tree = tree.predict(X_test)
y_train_tree = tree.predict(X_train)
```

Performance Analysis:

```
# Random Forest model
from sklearn.ensemble import RandomForestClassifier

# instantiate the model
forest = RandomForestClassifier(max_depth=5)

# fit the model
forest.fit(X_train, y_train)
```

```
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=5, max_features='auto',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=100,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False)
```

```
#predicting the target value from the model for the samples
y_test_forest = forest.predict(X_test)
y_train_forest = forest.predict(X_train)
```

Performance Evaluation:

```
#computing the accuracy of the model performance
acc_train_forest = accuracy_score(y_train,y_train_forest)
acc_test_forest = accuracy_score(y_test,y_test_forest)

print("Random forest: Accuracy on training Data: {:.3f}".format(acc_train_forest))
print("Random forest: Accuracy on test Data: {:.3f}".format(acc_test_forest))
```

```
Random forest: Accuracy on training Data: 0.818
Random forest: Accuracy on test Data: 0.815
```

Comprision of Models:

To compare the models performance, a dataframe is created. The columns c

```
#creating dataframe
results = pd.DataFrame({ 'ML Model': ML_Model,
    'Train Accuracy': acc_train,
    'Test Accuracy': acc_test})
results
```

40]:

	ML Model	Train Accuracy	Test Accuracy
0	Decision Tree	0.813	0.816
1	Random Forest	0.818	0.814
2	XGBoost	0.868	0.859
3	SVM	0.803	0.800

```
#Sorting the datafram on accuracy
results.sort_values(by=['Test Accuracy', 'Train Accuracy']
```

41]:

	ML Model	Train Accuracy	Test Accuracy
2	XGBoost	0.868	0.859
0	Decision Tree	0.813	0.816
1	Random Forest	0.818	0.814
3	SVM	0.803	0.800

Conclusion

This project started with Literature survey done via gathering information from different IEEE papers, patented documents and reputed Websites. Books from Orally publication were also used as a guide. Also, popular application were listed against their features making it clearer to compare and contrast applications with each other along with application proposed in this project. Background Theory of all resources including technology used, Framework selected, IDE's worked upon and engines applied were extensively elaborated so that these theory can be applied effectively and the reason for their use/ application can be well understood. Later, all objectives were listed done after declaring title and Aim of the project and methods and mythologies in order to complete the bulleted objectives were well tabulated. From objectives, Functional Requirement was extracted and were well segregated for sequential completion of project. Later, diagrams including sequence and architecture diagram were created giving complete view of the project from different perspective. Implementation of the project was displayed via displaying code written in python programming language interacting with each other to create the application. Testing was done for all functionalities and were found to be working successfully. Later in result section, all screenshot of application in different states were taken in order to demonstrate the end product of this project. Each screenshot was explained with its importance as a view for the application. Performance analysis was done in order to give numerical value to performance of the application clearly stating the

advancement in application proposed in this project compared to other applications present in the market. Later project cost estimation was done in order to know the financial asset required to rebuild this project. The entire project was

concluded with s suitable conclusion and its scope in near future.

Suggestion for future work

Innovation is a never-ending process, hence bring new innovation and extension of this project is always possible. Now we may be lagged of some features. In the future they may be solved and improve the project performance and accuracy.

References:

1. Gunter Ollmann, "The Phishing Guide Understanding & Preventing Phishing Attacks", IBMInternet Security Systems, 2007.

2. Hong J., Kim T., Liu J., Park N., Kim SW, "Phishing URL Detection with Lexical Features and Blacklisted Domains", *Autonomous Secure Cyber Systems*. Springer, 10.1007/978-3030-334321_12.