IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 7.39 ⁂ Vol. 11, Issue 2, February 2022

DOI: 10.17148/IJARCCE.2022.11245

Phishing Website Detection using Machine Learning

Gayathri V[1]*, Dr. Malatesh S H[2]

[1]Asst Professor , Nrupatunga University, Bengaluru

[2]Professor , M S Engineering College, Bengaluru

Abstract: Phishing attack is one of the commonly known attack where the information from the internet users is stolen by the intruder. The internet users are losses their sensitive information such as Protected passwords, personal information and their transactions to the intruders. The Phishing attack is normally carried by the attackers where the legitimate frequently used websites are manipulated and masked to gather the personal information of the users. The Intruders use the personal information and can manipulate the transactions and get definite from them. From the literature there are various anti-Phishing websites by the various authors. Some of the techniques are Blacklist or Whitelist and heuristic and visual similarity-based methods. In spite of the users using these techniques most of the users are getting attacked by the intruders by means of Phishing to gather their sensitive information. A novel Machine Learning based classification algorithm has been proposed in this paper which uses heuristic features where feature selection can be extracted from the attributes such as Uniform Resource Locator, Source Code, Session, Type of security involve, Protocol used, type of website. The proposed model has been evaluated using five machine learning algorithms such as random forest, Decision Tree, Logistic regression. Out of these models, the random forest algorithm performs better with attack detection accuracy of 92%. More over the Random Forest Model uses orthogonal and oblique classifiers to select the best classifiers for accurate detection of Phishing attacks in the websites.

Keywords: Phishing attack; Personal Machine Learning; Classification Algorithms; Cyber Security.

1. INTRODUCTION

The COVID-19 pandemic has boosted the use of technology in every sector, resulting in shifting of activities like organizing official meetings, attending classes, shopping, payments, etc. from physical to online space. The Internet has become an effective channel for social interactions nowadays. Peoples' immense dependence on digital platform opens doors for fraud. This means more opportunities for phishers to carry out attacks impacting the victim financially, psychologically & professionally. The phishing website has evolved as a major cyber security threat in recent times. The phishing websites host spam, malware, ransom ware, drive-by exploits, etc. Main aim of the attacker is to steal banks account credentials. A phishing website many times look like a very popular website and lure an unsuspecting user to fall victim to the trap. The victim of the scams incurs a monetary loss, loss of private information and loss of reputation. Phishing attacks are becoming successful because lack of user awareness. Since phishing attack exploits the weaknesses found in users, it is very difficult to mitigate them but it is very important to enhance phishing detection techniques.

The general method to detect phishing websites by updating blacklisted URLs, Internet Protocol (IP) to the antivirus database which is also known as "blacklist" method. To evade blacklists attackers uses creative techniques to fool users by modifying the URL to appear legitimate via obfuscation and many other simple techniques

II.RELATED WORK

Our work in this thesis focuses mainly on detecting phishing websites with machine learning. There has been quite some effort regarding similar topics such as malicious domain blacklisting and email spam filtering. Furthermore, it is increasingly popular to utilize machine learning in these areas. Existing malicious websites detection approaches can be mainly divided into two categories based on the features leveraged: static feature-based approaches and dynamic feature-based approaches. Static feature-based approaches rely on features extracted from the URL, page content, HTML DOM structure, domain-based information (such as WHOIS and DNS records) and so on. Alternatively, dynamic feature-based solutions primarily focus on analysing behaviours captured when the page is loaded and rendered, or investigating system logs when some scripts are executed. In this thesis, we concentrate on exploiting static features.

III.PROPOSED SYSTEM

In our proposed model for detecting phishing websites, we are going to develop machine learning based model by classification technique to enhance the detection accuracy. We will add comparative analysis between various classification algorithms and feature selection scenarios.
The proposed approach work starts by
1. Finding the correct dataset with complete features.
2. Secondly, we investigate previous work of different authors on the datasets and several classification algorithm techniques applied on the field.
3. Subsequently, we analyse the accuracy of different phishing website detection techniques applied on our dataset.
4. We choose several main features which can be further improved using DNN alongside with LSTM. By doing so, we can reduce time and space to apply our model.
5. We analyse and verify our model and results by comparing with the results of several classification algorithms on the selected features.

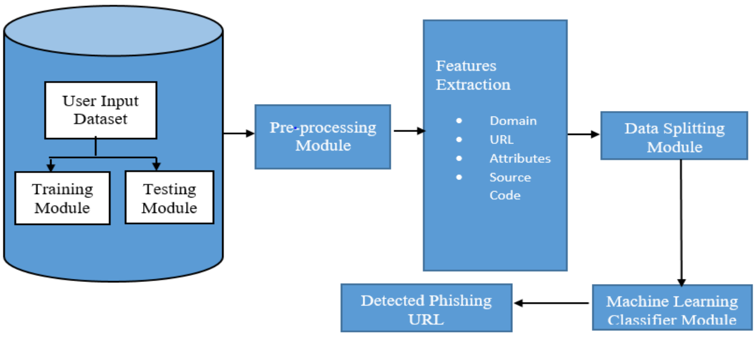


Fig III: Proposed Model to detect Phishing Attack

IV. DATASET

URLs of benign websites were collected from www.alexa.com and The URLs of phishing websites were collected from www.phishtank.com. The data set consists of total 36,711 URLs which include 17058 benign URLs and 19653 phishing URLs. Benign URLs are labelled as "0" and phishing URLs are labelled as "1".

V.FEATURE EXTRACTION

We have implemented python program to extract features from URL. Below are the features that we have extracted for detection of phishing URLs.
1) **Presence of IP address in URL:** If IP address present in URL, then the feature is set to 1 else set to 0. Most of the benign sites do not use IP address as an URLto download a webpage. Use of IP address in URL indicates that attacker is trying to steal sensitive information.
2) **Presence of @ symbol in URL**: If @ symbol presentin URL then the feature is set to 1 else set to 0. Phishers add special symbol @ in the URL leads the browser to ignore everything preceding the "@" symbol and the realaddress often follows the "@" symbol [4].
3) **Number of dots in Hostname**: Phishing URLs have many dots in URL. For example, http://shop.fun.amazon.phishing.com, in this URL phishing.com is an actual domain name, whereas use of "amazon" word is to trick users to click on it. Average number of dots in benign URLs is 3. If the number of dots in URLs is more than 3 then the feature is set to 1 else to 0.
4) **Prefix or Suffix separated by (-) to domain:** If domain name separated by dash (-) symbol then featureis set to 1 else to 0. The dash symbol is rarely used in legitimate URLs. Phishers add dash symbol (-) to the domain name so that users feel that they are dealing with a legitimate webpage. For example, Actual site ishttp://www.onlineamazon.com but phisher can create another fake website like http://www.online-amazon.comto confuse the innocent users.
5) **URL redirection:** If "//" present in URL path then feature is set to 1 else to 0. The existence of "//" within the URL path means that the user will be redirected to another website.
6) **HTTPS token in URL:** If HTTPS token present in URL, then the feature is set to 1 else to 0. Phishers may add the "HTTPS" token to the domain part of a URL in order to trick users. For example, http://https-www- paypal-it-mpp-home.soft-hair.com .

DOI: 10.17148/IJARCCE.2022.11245

7) **Information submission to Email:** Phisher might use “mail ()” or “mailto:” functions to redirect the user’s information to his personal email. If such functions are present in the URL, then feature is set to 1 else to 0.

8) **URL Shortening Services “TinyURL”:** TinyURL service allows phisher to hide long phishing URL bymaking it short. The goal is to redirect user to phishing websites. If the URL is crafted using shortening services (like bit.ly) then feature is set to 1 else 0

9) **Length of Host name:** Average length of the benign URLs is found to be a 25, If URL’s length is greater than 25 then the feature is set to 1 else to 0

10) **Presence of sensitive words in URL:** Phishing sitesuse sensitive words in its URL so that users feel that they are dealing with a legitimate webpage. Below are the words that found in many phishing URLs :- 'confirm', 'account', 'banking', 'secure', 'ebyisapi', 'webscr', 'signin', 'mail', 'install', 'toolbar', 'backup', 'paypal', 'password', 'username', etc;

11) **Number of slash in URL:** The number of slashes in benign URLs is found to be a 5; if number of slashes in URL is greater than 5 then the feature is set to 1 else to 0.

12) **Presence of Unicode in URL:** Phishers can make a use of Unicode characters in URL to trick users to click on it. For example the domain “xn--80ak6aa92e.com” is equivalent to "apple.com". Visible URL to user is "apple.com" but after clicking on this URL, user will visit to “xn--80ak6aa92e.com” which is a phishing site.

13) **Age of SSL Certificate:** The existence of HTTPS is very important in giving the impression of website legitimacy [4]. But minimum age of the SSL certificate of benign website is between 1 year to 2 year.

14) **URL of Anchor:** We have extracted this feature by crawling the source code oh the URL. URL of the anchoris defined by <a> tag. If the <a> tag has a maximum number of hyperlinks which are from the other domain then the feature is set to 1 else to 0.

15) **IFRAME:** We have extracted this feature by crawling the source code of the URL. This tag is used to add another web page into existing main webpage. Phishers can make use of the “iframe” tag and make it invisible i.e. without frame borders [4]. Since border of inserted webpage is invisible, user seems that the inserted web page is also the part of the main web page and can enter sensitive information.

16) **Website Rank:** We extracted the rank of websites and compare it with the first One hundred thousand websites of Alexa database. If rank of the website is greater than 10,0000 then feature is set to 1 else to 0

17)

VI.MACHINE LEARNING ALGORITHM

From the dataset created we can learn that this is a supervised machine learning task. There are two types of supervised machine learning tasks. They are Classification and Regression. Our project comes under classification as the input URL is classified as Phishing (1) and Legitimate (0).

So, below are the best classification supervised machine learning models which we are going to use to use to train our ML model.

- Logistic Regression
- Decision Tree Classifier
- Random Forest Model

6.1 Logistic Regression:

The logistic regression is a kind of predictive analysis were based on the attributes the phishing URLs can be detected. In logistic regression the input is given as training data and testing data. Based on the given input logistic regression is computed by using the regression function called sigmoid function with the computed sigmoid function the relationship between training data and testing data is calculated. Based on the relation the objects are categorized. If the patterns in the attributes of the training data and testing data are same, then the URLs are considered as phishing URLs else other URLs are considered as Legitimate URLs.

- **Logistic Regression: Accuracy on test Data: 0.918**

6.2 Decision Tree Classifier

The next category of machine learning classifier is decision tree algorithm. In decision tree the attributes with high information gain considered as different set of attributes where the certain decision can be obtain from the attributes of high information gain. In decision tree algorithm, the various phishing attributes with high information gain are compared with each other, the phishing attributes with high impact are considered as Phishing URLs and rest of the attributes are considered as legitimate URLs.

- **Decision tree: Accuracy on test Data: 0.916**

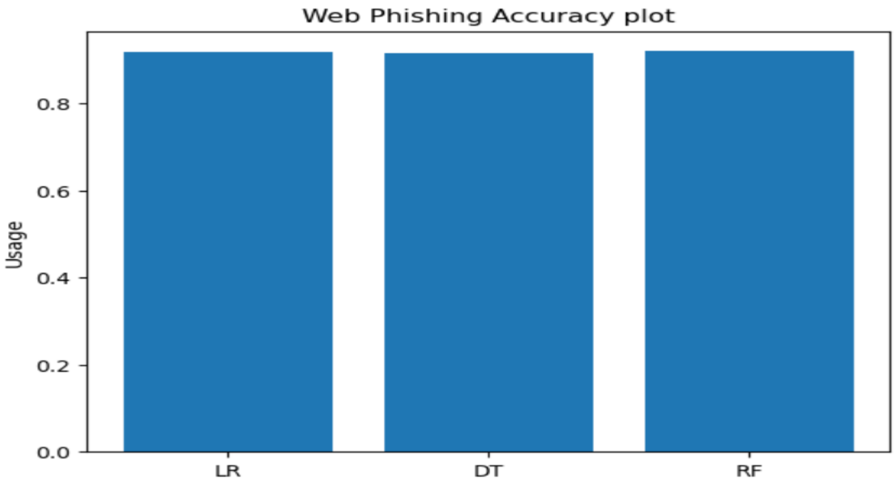
6.3 Random Forest

The next category of machine learning is random forest algorithm. The main aim of the random forest is to detect the phishing URLs from the legitimate URLs. Random forest is widely used ensemble learning methods and works by combination of all their output and predicts the best output among the test data.

- **Random forest: Accuracy on test Data: 0.920**

DOI: 10.17148/IJARCCE.2022.11245

The below graph shows the peformance evaluation of machine learning with test accuracy plotted.



Graph Usage Vs Ml model (LR, DT, RF)

As we tested, Random Forest give the better accuracy result by comparing with Decision Tree Classifier, Logistic Regression.

VII.IMPLEMENTATION

Implementation is the stage of the project where the theoretical design is turned out into a working system. Implementation is the carrying out, execution, or practice of a plan, a method, or any design, idea, model, specification, standard or policy for doing something.The nature of the implementation processes will depend on the type and size of the project. Thus, it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective. To implement a system successfully, a large number of inter-related tasks need to be carried out in an appropriate sequence. The implementation stage involves careful planning, investigation of the existing system and its constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

Implementation Part Consist of some Modules, They are as follows.

- Data Collection.
- Data Pre-processing.
- Data Splitting.
- Data Modelling.
- Data Evaluation
- Data Deploying.

Data Collection.

Data collection Model includes Loading the Data, Familiarizing with Data, and Visualizing the Data. Data Collection is the understanding the dataset is the initial part of our proposed model. Our dataset was collected from UCI and it has 11055 lists of websites.

Data Loading includes the features are extracted and store in the csv file. The working of this can be seen in the 'Phishing Website Detection Feature Extraction.ipynb' file. The resulted csv file is uploaded to the notebook and stored in the data frame, and Familiarizing with data includes a few data frame methods which are used to look into the data and its features, Visualizing the date consists of few plots and graphs are displayed to find how the data is distributed and the how features are related to each other.

```
#importing basic packages
 import warnings
 warnings.filterwarnings("ignore")
 import pandas as pd
 import numpy as np
#Loading the data
 data = pd.read_csv('/content/dataset.csv')
```

```
data.head()
```

Data Preprocessing

Here, we clean the data by applying data preprocesssing techniques and transform the data to use it in the models.

```
data.isna().sum()
data.shape
data.columns
from collections import Counter

classes = Counter(data['Result'].values)
classes.most_common()

class_dist = pd.DataFrame(classes.most_common(), columns=['Class', 'Num_Observations'])
class_dist

import matplotlib.pyplot as plt
%matplotlib inline
plt.style.use('ggplot')
subplot = class_dist.groupby('Class')['Num_Observations'].sum().plot(kind='barh', width=0.2, figsize=(10,8))
subplot.set_title('Class distribution of the websites', fontsize = 15)
subplot.set_xlabel('Number of Observations', fontsize = 14)
subplot.set_ylabel('Class', fontsize = 14)

for i in subplot.patches:
    subplot.text(i.get_width()+0.1, i.get_y()+0.1, \
            str(i.get_width()), fontsize=11)

data.describe().T
#Information about the dataset
data.info()
```

Data Splitting

```
# Splitting the dataset into train and test sets: 80-20 split
from sklearn.model_selection import train_test_split
X = data.iloc[:,0:30].values.astype(int)
y = data.iloc[:,30].values.astype(int)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=np.random.seed(7))
X_train.shape, X_test.shape, y_train.shape,y_test.shape
```

Data Modelling

Random Forest

Random forests for regression and classification are currently among the most widely used machine learning methods. A random forest is essentially a collection of decision trees, where each tree is slightly different from the others. The idea behind random forests is that each tree might do a relatively good job of predicting, but will likely overfit on part of the data.

If we build many trees, all of which work well and overfit in different ways, we can reduce the amount of overfitting by averaging their results. To build a random forest model, you need to decide on the number of trees to build (the estimator's parameter of RandomForestRegressor or RandomForestClassifier). They are very powerful, often work well without heavy tuning of the parameters, and don't require scaling of the data.

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

# instantiate the model
forest = RandomForestClassifier(max_depth=5,random_state=0)

# fit the model
```

```
forest.fit(X_train, y_train)
#predicting the target value from the model for the samples
y_test_forest = forest.predict(X_test)
acc_test_forest = accuracy_score(y_test,y_test_forest)
print("Random forest: Accuracy on test Data: {:.3f}".format(acc_test_forest))
```

Linear Regression Model

Linear regression may be defined as the statistical model that analyzes the linear relationship between a dependent variable with given set of independent variables. Linear relationship between variables means that when the value of one or more independent variables will change (increase or decrease), the value of dependent variable will also change accordingly (increase or decrease).

```
# Random Linear Regression model
from sklearn.linear_model import LogisticRegression

# instantiate the model
lr = LogisticRegression()

# fit the model
lr.fit(X_train, y_train)
y_test_lr = lr.predict(X_test)
acc_test_lr = accuracy_score(y_test,y_test_lr)
print("Logistic Regression: Accuracy on test Data: {:.3f}".format

import matplotlib.pyplot as plt; plt.rcdefaults()
import numpy as np
import matplotlib.pyplot as plt

objects = ['LR','DT','RF']
y_pos = np.arange(len(objects))
performance = [acc_test_lr, acc_test_dt, acc_test_forest]

plt.bar(y_pos, performance, align='center', alpha=1)
plt.xticks(y_pos, objects)
plt.ylabel('Usage')
plt.title('Web Phishing Accuracy plot')
plt.show( )
```

Data Evaluation

This Model includes comparing all the Models performance, a data frame is created. The columns of this data frame are the lists created to store the results of the model.

Deploying

This Model Species the Result which Model provides good performance and gives the Best Accuracy.

The below table shows test accuracy of machine learning model.

SL. N	ML Model	Test Accuracy
1	Logistic Regression	0.918
2	Decision Tree	0.916
3	Random Forest	0.920

Test Accuracy of Ml model

Random Forest give the better accuracy result by comparing with Decision Tree Classifier, Logistic Regression.

DOI: 10.17148/IJARCCE.2022.11245

CONCLUSION

This paper aims to enhance detection method to detect phishing websites using machine learning technology.

➢ In this paper, this survey presented various algorithms and approaches to detect phishing websites by several researchers in Machine Learning.

➢ On reviewing the papers, we came to a conclusion that most of the work done by using familiar machine learning algorithms like Decision Tree Classifier, Random Forest, logistic Regression.

➢ For the detection of various phishing websites Random Forest Classifier which is used to obtain accuracy.

➢ The algorithm of classification analysis the performance based on the literature review is proof to give 92% accuracy.

➢ Hence Random Forest is selected for the analysis which performs better when compared to the Decision Tree and Logistic Regression algorithm. Random Forest has an accuracy of around 92 percentage and also it is time saving when compared to Decision Tree algorithm and Logistic Regression.

➢ Decision Tree algorithm and Logistic Regression takes up a time more than five minutes to result the output whereas in Random Forest it only takes very few seconds.

➢ Thus, it can be summarized that better algorithm is chosen and the experimentally successful technique in detecting phishing website.

FUTURE SCOPE

➢ In future ways we can test this phishing websites through many ways according to our technology development.

➢ If we get structured dataset of phishing, we can perform phishing detection much faster than any other technique.

➢ Also, we can use a combination of any other two or more classifier to get maximum accuracy.

➢ We plan to explore various phishing techniques that uses Lexical features, Network based features, Content based features, Webpage based features and HTML and JavaScript features of web pages which can improve the performance of the system.

➢ In particular, we extract features from URLs and pass it through the various classifiers.

Therefore, the future works can be to fix the antivirus also into the tool in which the user will be comfort to access all websites and be secure through it.

REFERENCE

[1] Moitrayee Chatterjee and Akbar Siami Namin; “Detecting Phishing Websites through Deep Reinforcement Learning ”, IEEE, 2019

[2] Ishita Saha, Dhiman Sarma, Rana Joyti Chakma , Mohammad Nazmul Alam, Asma Sultana and Sohrab Hossain; “Phishing Attacks Detection using Deep Learning Approach”, IEEE, 2020

[3] Mahdieh Zabihimayvan and Derek Doran; “Fuzzy Rough Set Feature Selection to Enhance Phishing Attack Detection” IEEE, 2019

[4] Erzhou zhu , Yuyang Chen, Chengcheng Ye and Xuejun Li;“An Effective Phishing Websites Detection Model Based on Optimal Feature Selection and Neural Network ”, IEEE, 2019

[5] Yasin Sonmez, Turker Tuncer, Huseyin Gokal and Engin Avc; “Phishing Web Sites Features Classification Based on Extreme Learning Machine ”, IEEE, 2018.

[6] https://resources.infosecinstitute.com/category/enterprise/phishing/the-phishing-landscape/phishing-data-attack-statistics/#gref

[7] Mahmoud Khonji, Youssef Iraqi, "Phishing Detection: A Literature Survey IEEE, and Andrew Jones, 2013

[8] Mohammad R., Thabtah F. McCluskey L., (2015)Phishing websites dataset. Available:https://archive.ics.uci.edu/ml/datasets/Phishing+Websites Accessed January 2016

[9] http://dataaspirant.com/2017/01/30/how-decision-tree-algorithm-works/http://dataaspirant.com/2017/05/22/random-forest- algorithm-machine-learning.

[10] Gunter Ollmann, “The Phishing Guide Understanding & Preventing Phishing Attacks”, IBMInternet Security Systems, 2007.