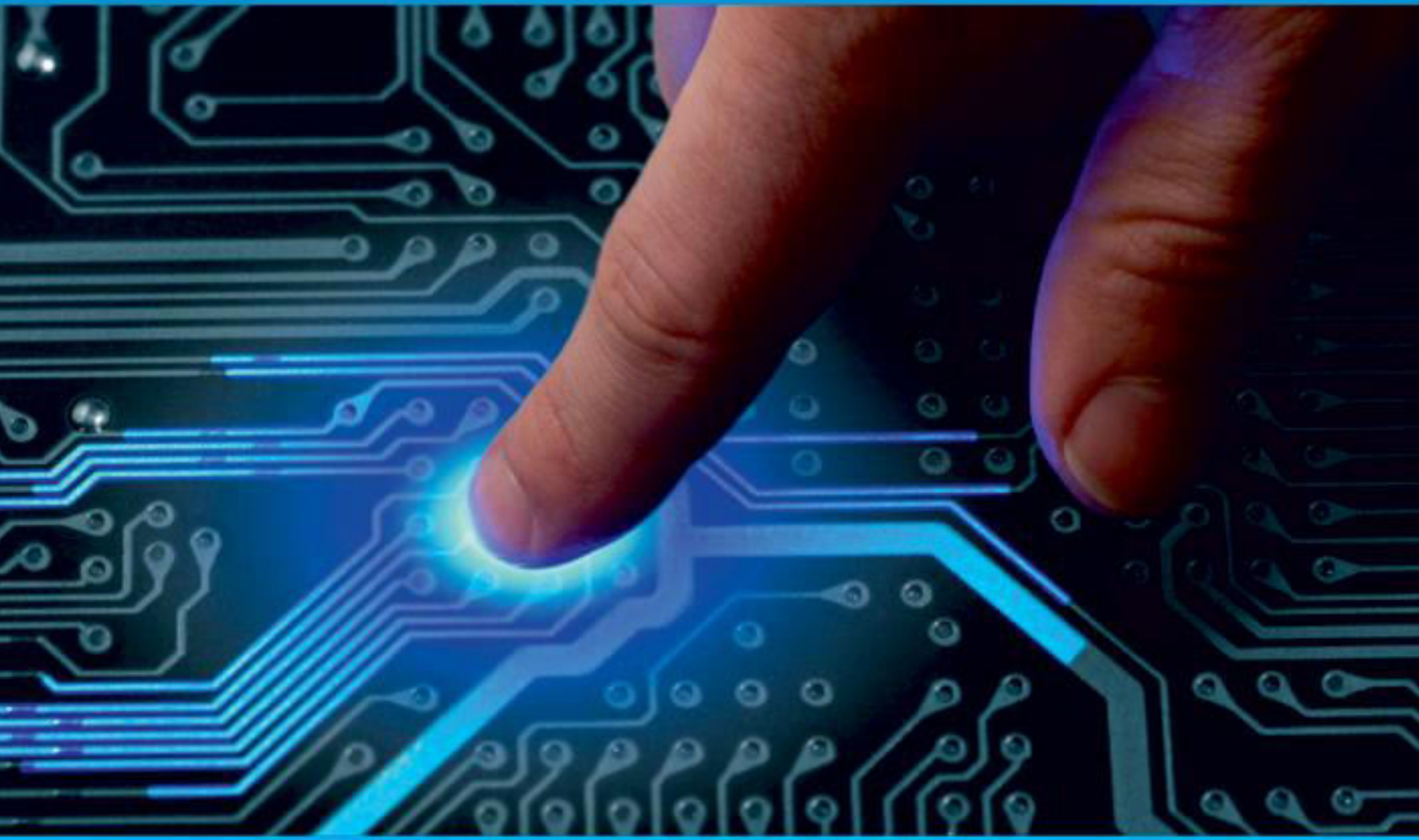
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 10, Issue 6, June 2022

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.165

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 6, June 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1006142|

Classification and Detection of Phishing URLs using Machine Learning Approach

Nandeesh H D, Prathyusha V Raikar, Vaishnavi Srinath, Chinmayi P, Dhanush R

Department of Computer Science, JSS Science and Technology University, Mysore, India

ABSTRACT: Phishing, a form of cyber-attack, with an adverse effect where the user is directed to fake websites and duped to reveal their sensitive and personal information which includes passwords of accounts, bank details, atm pin-card details etc. Hence, protecting sensitive information from web phishing is difficult.We perform detailed literature survey and proposed new approach to detect phishing URLs by feature extraction and machine learning algorithms. A Voting Classifier is used that trains on an ensemble of numerous models ,i.e, Random forest, Adaboost and Gradient Boosting and predicts an output based on the average of probability given to that class.

1. INTRODUCTION

Phishing is a type of social engineering attack in which the attacker sends a fake mail or sms, through which the victim will make some sort of sensitive information available for the attacker to further break into the victim's system and cause potential damage. The fraudulent mails usually redirect the victim to some malicious websites. Phishing attacks have become increasingly sophisticated, allowing the attacker to observe everything while the victim is navigating the site, and transverse any additional security boundaries with the victim.

The present phishing detection techniques suffer low detection accuracy. One of the most common and traditional detection techniques used is the Blacklist-based method and then the visual similarity based approach and heuristic based which are inefficient in responding to phishing attacks. Hence Machine learning has been a major breakthrough which can remove all existing drawbacks, But there is a scope of improvement with greater levels of accuracy.

II. LITERATURE SURVEY

We have gone through several papers related to our project domain. The summary of few of them are as follows:
Mohammad et al [1] proposed a dataset based on 30 features, with each feature being categorical and classified as legitimate, suspicious or phishing. Each datapoint is then classified as legitimate or phishing. The dataset has 11055 data points with 6157 legitimate URLs and 4898 phishing URLs. The dataset was accompanied by a set of rules to categorise features for any new URL.
Himanshi et al[2] presented a machine learning based approach combining 25 features to attain the best results from Dataset provided by UCI Machine Learning repository.SVM achieved an accuracy of 96.29%. They also provided the output as a user-friendly web platform.
IshantTyagi et al[3] Cleaning of the Dataset is done using Variability Inflation Factor and Principal Component Analysis and K-Nearest-Neighbors. In experiments several well-known classification algorithms were tested. This was done using the open-source programming language R. The algorithms have been ranked based on their overall performance and managed an accuracy of 98.4% through random forest.
Saeed et al[4] compares the predictive accuracy of several machine learning methods including Logistic Regression (LR), Classification and Regression Trees (CART), Bayesian Additive Regression Trees (BART), Support Vector Machines (SVM), Random Forests (RF), and Neural Networks (NNet) for predicting phishing emails. A data set of 2889 phishing and legitimate emails is used in the comparative study. In addition, 43 features are used to train and test the classifiers.
Marchal et al[5] used Phishtank and OpenPhish as the sources for the phishing dataset. These datasets include columns such as phishing URL, target brand name, IP etc. In this paper, machine learning algorithms used are - J48, Support Vector Machine (SVM), and Logistic Regression (LR). J48 has the highest accuracy with accuracy of 96.96%.
Sohail et al[6] presents a comprehensive analysis of various machine learning algorithms to evaluate their performances over multiple datasets. The statistical results indicate that random forest and artificial neural networks outperform other classification algorithms, achieving over 97% accuracy using the identified features.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 6, June 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1006142|

Tharani et al[7] works on two phases. Phase 1 identifies the topmost features that are employed by cyber-criminals to mimic URLs. In phase 2, 48 features have been considered and performed classification for best 20, 15, and 10 features.For the top 10 features the KNN classifier performed better than the other two classifiers and When the number of features increase to 15 and 20 Linear SVM and KNN, are providing best accuracy rates respectively.

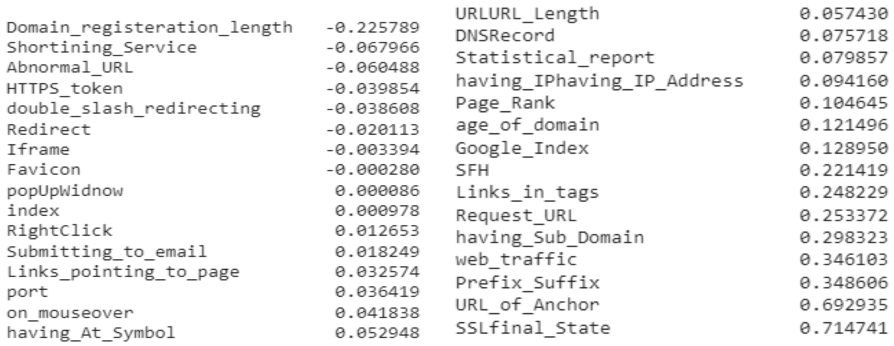
III. PROPOSED METHODOLOGY AND DISCUSSION

3.1 Dataset

We have collected a structured dataset containing more than 10,000 URLs from UCI machine learning repository. The Phishing URLs are labelled as “-1” and legitimate URLs are labelled as “1” and we have used a 70:30 ratio as training and testing data split.

3.2 Feature Selection

We have removed the features, i.e, Redirect, Favicon, Iframe, RightClick, Submitting_to_email, popUpWidnow which are having correlation values in the range -0.03 to + 0.03 and not contributing to either positive correlation or negative correlation.



```
Domain_registeration_length    -0.225789
Shortining_Service             -0.067966
Abnormal_URL                   -0.060488
HTTPS_token                    -0.039854
double_slash_redirecting       -0.038608
Redirect                       -0.020113
Iframe                         -0.003394
Favicon                        -0.000280
popUpWidnow                     0.000086
index                           0.000978
RightClick                      0.012653
Submitting_to_email             0.018249
Links_pointing_to_page          0.032574
port                            0.036419
on_mouseover                    0.041838
having_At_Symbol                0.052948
URLURL_Length                   0.057430
DNSRecord                       0.075718
Statistical_report              0.079857
having_IPhaving_IP_Address      0.094160
Page_Rank                       0.104645
age_of_domain                   0.121496
Google_Index                    0.128950
SFH                             0.221419
Links_in_tags                   0.248229
Request_URL                     0.253372
having_Sub_Domain               0.298323
web_traffic                     0.346103
Prefix_Suffix                   0.348606
URL_of_Anchor                   0.692935
SSLfinal_State                  0.714741
```

Table 1 . Correlation values of each feature

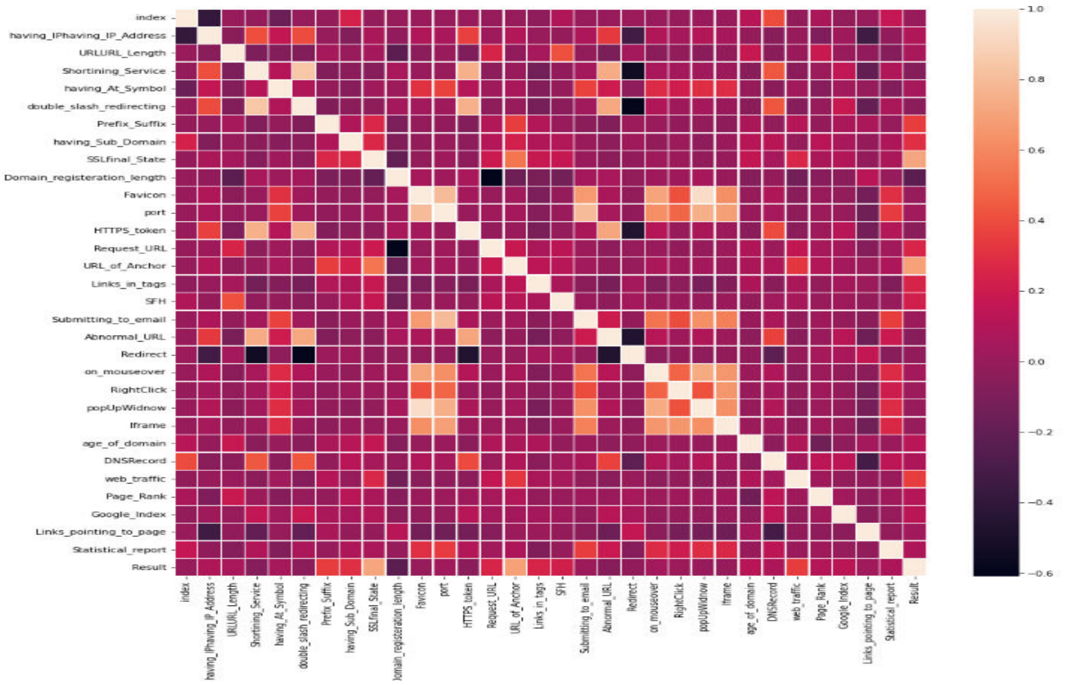


Fig 1. Correlation matrix for features

3.3 Methodology

3.3.1 Model details

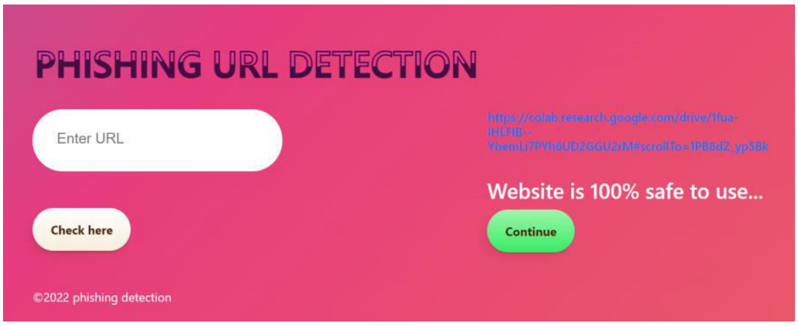
1.Logistic Regression: It is a statistical model that uses a logistic function to classify the data points
2.K-Nearest Neighbour: KNN calculates the nearest K neighbours for each data point and returns the majority label among them.The hyperparameters used are n_neighbours as 3.
3.Random Forest Classifier: It is an ensemble Classification model which takes average results from multiple decision trees and optimally predicts. The number of estimators taken is 35.
4.Decision Tree: It creates a classification model that learns by creating decision boundaries. The maximum depth used is 15.
5.Support Vector Machine: SVM classifies the given labelled training data by creating an optimal hyperplane for classification. The kernel used was linear.
6. Naives Bayes: It is a probabilistic model that assumes the features to be independent of each order.
7.Gradient Boosting:In gradient boosting, each predictor corrects its predecessor's error, here many weak learners come up with one strong learner.
8.XGBoost: XGBoost is based on decision trees that use a gradient boosting framework for classification and designed for speed and performance.
9.Adaboost: In Adaboost ,the weights of the training instances are not tweaked, instead, each predictor is trained using the residual errors of the predecessor as labels.

3.3.2 Model building

We have implemented all the 9 Machine learning Models in the google collaborator. In order to select the best parameters for the model (i.e, learning rate , maximum depth) we have used 2 techniques , plotting the values and Grid search technique.
We have chosen voting classifier as an ensemble model embedded with 3 classifier random forest which in turn is an ensemble model under bagging technique and Gradient Boosting and Adaboosting which are Boosting algorithms of ensemble technique , since we have made use of bagging and boosting inside the model as individual learning models , we chose voting classifier as a final model to classify the data on top of these 3 learning models. The final ensemble model is saved in the '.sav' file with the help of pickle python module .

3.4 Front end implementation

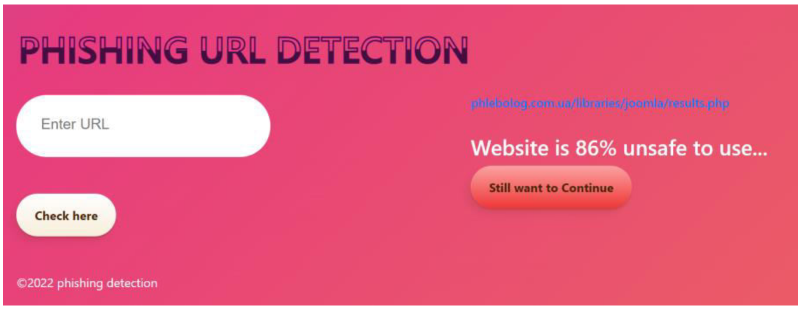
A form is displayed to enter any real time URL and the 'check here' button will submit the form to the backend of the project where data extraction and prediction takes place. If the entered URL is valid , we will be shown that the website is safe with class probability and a 'continue' green button which will route the user to the input url he had entered. If the entered URL is invalid, we will be shown that the website is unsafe with class probability and 'still want to continue' red button which will also route the user to the input url he had entered.



https://colab.research.google.com/drive/1fua-IHLFIB--YbemLi7PYh6UD2GGU2rM#scrollTo=1PB8dZ_yp5Bk

Website is 100% safe to use...

Fig 2 . Website showing URL entered is legitimate.


PHISHING URL DETECTION

phlebolog.com.ua/libraries/joomla/results.php

Website is 86% unsafe to use...

Fig 3 . The website showing the URL entered is Phishing.

3.5 Back end implementation

We have used a flask application to take the input from the front end and later fed onto feature extraction where all the 24 features are extracted. We have used various python modules such as ipaddress, beautifulSoup , Alexa and WHOIS database to extract the desired feature.

Srl	Feature Name	Feature Description
1	having_IP_Address	Using an IP address in the domain name of the URL.
2	URL_Length	Phishers can use long URL to hide the doubtful part in the address bar
3	Shortining_Service	URL shortening service is a third-party website that converts that long URL to a short, case-sensitive alphanumeric code.
4	having_At_Symbol	The ‘‘@’’ symbol leads the browser to ignore everything prior to it and redirects the user to the link typed after it.
5	double_slash_redirecting	The existence of “//” within the URL path means that the user will be redirected to another website.
6	Prefix_Suffix	Phishers try to scam users by reshaping the suspicious URL, so it looks legitimate. One technique used is adding a prefix or suffix to the legitimate URL. Thus, the user may not notice any difference.
7	having_Sub_Domain	Another technique used by phishers to scam users is by adding a subdomain to the URL so users may believe they are dealing with an authentic website.
8	SSL	final_State is a standard security technology for establishing an encrypted link between a server and a client.
9	Domain_registeration_length	Based on the fact that a phishing website lives for a short period of time, we believe that trustworthy domains are regularly paid for several years in advance.
10	port	This feature is useful in validating if a particular service such as HTTP is up or down on a specific server. In the aim of controlling intrusions, it is much better to merely open ports that you need.
11	HTTPS_token	IF The phishers may add the HTTPS token to the domain part of a URL in order to trick users.
12	Request_URL	If the objects are loaded from a domain other than the one typed in the URL address bar, the webpage is potentially suspicious.
13	URL_of_Anchor	Similar to the URL feature, but here the links within the webpage may point to a domain different from the domain typed in the URL address bar.

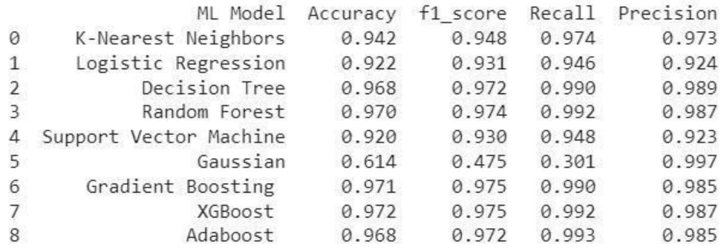
14	Links_in_tags	Links present in tags like META and SCRIPT are checked
15	SFH	Server Form Handlers containing an empty string or “about:blank” are considered doubtful because an action should be taken upon the submitted information.
16	Abnormal_URL	If the website identity does not match a record in the WHOIS database (WHOIS, 2011) the website is classified as phishy.
17	on_mouseover	Phishers often hide the suspicious link by showing a fake link on the status bar of the browser or by hiding the status bar itself. This can be achieved by tracking the mouse cursor and once the user arrives to the suspicious link the status bar content is changed
18	age_of_domain	Websites that have an online presence of less than 1 year, can be considered risky.
19	DNSRecord	An empty or missing DNS record of a website is classified as phishing.
20	web_traffic	Legitimate websites usually have high traffic since they are being visited regularly. Since phishing websites normally have a relatively short life; they have no web traffic or they have low ranking.
21	Page_Rank	PageRank is a value ranging from “0” to “1”. The greater the PageRank value the more important the webpage. It is found that 95% of phishing webpages have no PageRank and the remaining 5% of phishing webpages have a PageRank value up to “0.2”.
22	Google_Index	This feature examines whether a website is in Google’s index or not.
23	Links_pointing_to_page	More the number of links referring to a webpage , more is the website secure.It is found that 98% of phishing dataset items have no links pointing to them. On the other hand, legitimate websites have at least 2 external links pointing to them.
24	Statistical_report	formulate numerous statistical reports on phishing websites at every given period.

Table 2. Feature description

We use the same pickle module to load the save ‘.sav’ file to the flask app.Once the features are extracted , the model is fed with the extracted feature to predict the output and the output is rendered back to index.html to display in the front end with the help of javascript functions.

IV. RESULTS

For evaluating phishing classification performance we have used accuracy, recall, precision, F1 score, test time and train time of classifiers.Accuracy is the ratio of the number of correct predictions to the total number of input samples. Recall measures the percentage of phishing websites that the model manages to detect (models effectiveness). Precision measures the degree to which the phishing detected websites are indeed phishing (models safety). F1 score is the weighted harmonic mean of precision and recall.



```
                 ML Model  Accuracy  f1_score  Recall  Precision
0     K-Nearest Neighbors     0.942     0.948   0.974      0.973
1     Logistic Regression     0.922     0.931   0.946      0.924
2           Decision Tree     0.968     0.972   0.990      0.989
3           Random Forest     0.970     0.974   0.992      0.987
4  Support Vector Machine     0.920     0.930   0.948      0.923
5                Gaussian     0.614     0.475   0.301      0.997
6       Gradient Boosting     0.971     0.975   0.990      0.985
7                 XGBoost     0.972     0.975   0.992      0.987
8                Adaboost     0.968     0.972   0.993      0.985
```

Table 3 . Results of all classifier

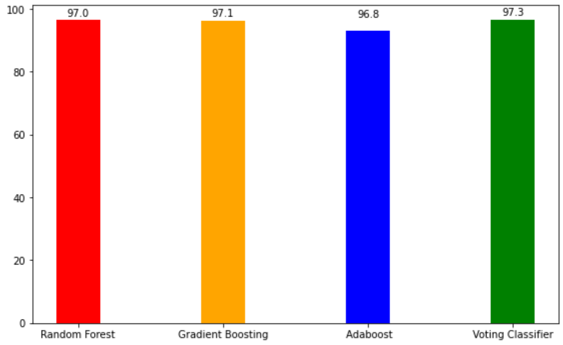


Fig 4. Comparison of voting classifier with 3 chosen classifier

V. CONCLUSION & FUTURE WORK

In this project, we investigated the problem of phishing URLs and adapted machine learning algorithms to mitigate the issue. We trained our dataset on various algorithms in 4 different ratio splits , i.e, 60:40 , 70:30, 80:20 and 90:10 , we observed that 70:30 is more accurate and gives maximum accuracy. We proposed an ensemble model called voting classifier on top of 3 chosen algorithms (Random forest, gradient boosting, adaboost). These 3 detectors slightly varied in their result , yet all of them scored less accuracy than the combined ensemble to seek its applicability to the phishing problem.we have used features from various domains spanning from URL to HTML tags of the web pages, from embedded URLs to favicon, and databases like WHOIS, alexa, pagerank and few more to check the traffic and status of the websites. Some of the prominent Future Works are,

Browser extension: Browser extension can make it more convenient and easier than a web platform.They help to prevent accidental land ups on phishing websites, by checking every URL which the browser tries to open , before actually allowing the user to land up on the pages.

Caching results in a database: Currently, for all queries, API hits and web scrapping is done every time an URL is entered. If some of the features and results are cached in a database, the query time can be reduced.

Parallel Feature extraction: For now, the feature extraction for an input URL is done sequentially one after the other and stored in an array, but this can be made parallel if more computation power is available.The scrapping using beautifulSoup and curl,WHOIS, alexa and other database lookups and string parsing can be done parallely with work-stealing.

REFERENCES

1. R. M. Mohammad, F. Thabtah, and L. McCluskey.” An assessment of features related to phishing websites using an automated technique”. In 2012 International Conference for Internet Technology and Secured Transactions, pages 492–497,2012.
2. Himanshi Mathur, Vanshika Goel, Vibhu Agrawal.”WhatAPhish: Detecting Phishing Websites “, Department of Computer Science Indraprastha Institute of Information Technology, Delhi
3. I. Tyagi, J. Shad, S. Sharma, S. Gaur and G. Kaur, "A Novel Machine Learning Approach to Detect Phishing Websites," 2018 5th International Conference on Signal Processing and Integrated Networks (SPIN), 2018, pp. 425-430, doi: 10.1109/SPIN.2018.8474040.
4. Abu-Nimeh, S., Nappa, D., Wang, X. and Nair, S., 2007. A comparison of machine learning techniques for phishing detection. Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit on - eCrime '07,.
5. S. Marchal, J. Francois, R. State, and T. Engel, “PhishScore: hacking phishers’ minds,” in proceedings of the 10th International Conference on Network and Service Management 2014 (CNSM 2014), vol. 11, no. 4, pp. 458-471, 2014.
6. Khan, S., Khan, W. and Hussain, A., 2020. Phishing Attacks and Websites Classification Using Machine Learning and Multiple Datasets (A Comparative Analysis). Intelligent Computing Methodologies, pp.301-313.
7. Tharani, J. & Arachchilage, Nalin. (2020). Understanding phishers' strategies of mimicking uniform resource locators to leverage phishing attacks: A machine learning approach. Security and Privacy. 3. 10.1002/spy2.120.

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

Scan to save the contact details

www.ijircce.com