PHISHING WEBSITE DETECTION BY MACHINE LEARNING TECHNIQUES

A PROJECT REPORT

Submitted by

Mohd. Umar (18BCE0196)

Manndeep Roy (19BEI0073)

Course Code: CSE 3021
Social Information Networks (EPJ)

S l o t -(B2)

Project Guide

Prof. Manjula R
Assistant Professor Sr.
School of Computer Science and Engineering

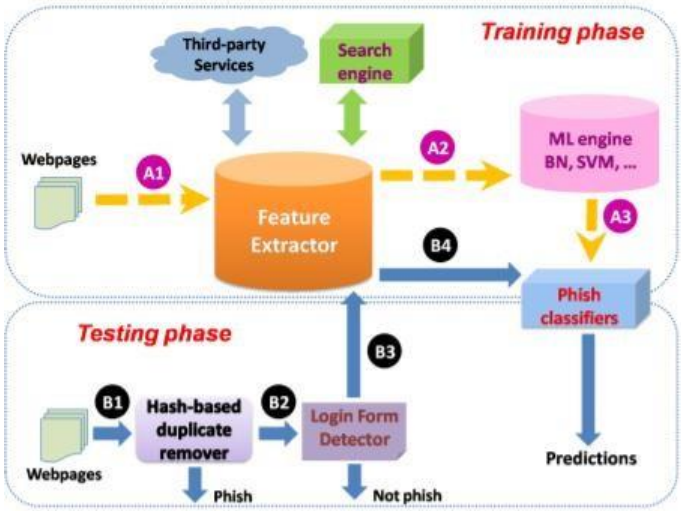
Fall Semester 2020 - 2021

ABSTRACT

Internet today has managed to globally connect millions of users around the world as a result of which there has been an increase in the dependency of the users to use this platform for browsing data, making online payments and downloading information. Cyber security refers to a body of technologies and processes to protect the programs and devices from unauthorized access, damage and attacks. The most commonly observed attacks under cybersecurity includes DOS attacks, Man-in-the middle attack, Phishing attack, SQL injection attacks etc. In the past few years, there has been an increase in the vulnerability levels of the users falling prey into loosing, their private and highly personal information. Nowadays, criminals make use of such techniques to deceive victims, attempting to obtain sensitive information such as the user's username, password, bank account and credit card details. The users are commonly exposed to attacks through spoofing emails, Illegal websites, malwares etc. A structured automated approach is essential towards handling complex and large amount of data. Machine learning has been proven to be the most popular and essential technique which can be adopted to solve this problem. The most commonly used machine learning techniques includes: Logistic regression, Support Vector Machine (SVM), Decision Tree and Neural Networks. In this paper, a set of machine learning and deep learning models are to be trained to predict phishing websites.

INTRODUCTION

Due to the involvement of the network, most of our daily life activities like shopping, banking are transferred to the cyberspace. The uncontrolled or unprotected network provides the platform for the various cyberattacks, which presents serious security vulnerabilities not only for networks but also for the standard computer users even for the experienced ones. Its, very important to prevent the users from these cyberattacks. One of the most common cyberattack is phishing website attack. A phishing website is a common social engineering method that mimics trustful uniform resource locators (URLs) and webpages. Through such attacks, the phisher targets naïve online users by tricking them into revealing confidential information, with the purpose of using it fraudulently. Due to the vulnerability of the end user, an attacker can even target some experienced users by using new techniques and before giving the sensitive information, he is believed that this page is legitimate. Therefore, software-based phishing detection systems are preferred as decision support systems for the user. In this project Machine learning algorithms techniques are adopted. The objective of this project is to train machine learning models and deep neural nets on the dataset created to predict phishing websites. Both phishing and benign URLs of websites are gathered to form a dataset and from them required URL and website content-based features are extracted. The performance level of each model is measured and compared. The Feature extraction will be done based on the address bar, domain based, HTML & Javascript based extraction before feeding into the algorithms. The Machine learning algorithms that will be used in this project are Decision Tree, Random Forest, Multilayer Perceptrons, XGBoost, Support Vector Machines. The models are going to be evaluated, and the considered metric is accuracy. The expected result of the algorithm performing better in this case is XGBoost as it has in-built L1 (Lasso Regression) and L2 (Ridge Regression) regularization which prevents the model from overfitting. Moreover, XGBoost has an in-built capability to handle missing values. When XGBoost encounters a missing value at a node, it tries both the left and right hand split and learns the way leading to higher loss for each node. The vulnerability in this project is the security of the website as cyberattacks can happen because less security Incorporated in the site.



LITERATURE SURVEY

1. "Machine learning based phishing detection from URLs Ozgur Koray Sahingoz , Ebubekir Buber , Onder Demir , Banu Diri".This paper proposes the implementation of a phishing detection system by using seven different machine learning algorithms, as Decision Trees, Ada boost, K-star, KNN (n = 3), Random Forest, SMO and Naive Bayes, and different number/types of features as NLP based features, word vectors, and hybrid features. It was clearly seen that the NLP based features displayed better performance than word vectors with the average rate of 10.86%. In addition it was seen that the use of NLP based features and word vectors together also increased the performance of the phishing detection system by a rate of 2.24% with respect to NLP based features and 13.14% with respect to word vectors. It has been noted that the adaptation of deep learning and parallel processing techniques coupled with a larger training dataset can further increase the efficiency of the system.

   Advantages
   1) It was seen that the use of NLP based features and word vectors together also increased the performance of the phishing detection system by a rate of 2.24% with respect to NLP based features and 13.14% with respect to word vectors.
   2) Use of Feature-Rich Classifiers
   3) Independence from Third-Party Services

2. "An Enhanced Classification Model to Detect Malicious URLs 1,* 1 1 Shweta Sankhwar, Dhirendra Pandey and R.A Khan ".This paper proposes a reliable mechanism i.e. Enhanced Malicious URLs Detection (EMUD) model to combat against numerous phishing URLs crafting tactics used to bypass the detection techniques. Supervised machine learning techniques (i.e. NB and SVM) to detect malicious URLs with the EMUD algorithm has been used with EMUD model. EMUD model has more effective detection capabilities as it includes more detection parameter (relevant URL heuristics) to catch and detect malicious URLs in comparison with other existing URL detection algorithms. It was concluded that EMUD model detects the phishing/ obfuscated URLs more accurately with SVM. EMUD model can be more effective by adding latest pertinent heuristics for zero-day phishing detection.

   Advantages
   1) The proposed EMUD model achieves 93.01% accuracy with 90% of True Positive (TPR) and 4.90% False Positive Rate (FPR)

   Drawbacks

   1) Use of NB classifiers lead to long processing times

3. Towards detection of phishing websites on client-side using machine learning based approach Ankit Kumar Jain · B. B. Gupta. This paper presented a novel approach for filtering phishing websites at client side where URL, hyperlink, CSS, login form, and identity features are used. The main contribution of this paper is the identification of various new client-side specific features that are previously not studied. The experimental results showed that the solution is very efficient as it has 99.39% true positive rate and only 1.25 %

false positive rate. The proposed approach also has good accuracy as compared to other existing anti-phishing solutions. The feature set of this phishing detection approach entirely depends on the URL and source code of the web-site, which can detect the webpages written in HTML code only. The paper further outlines that detecting the phishing websites in the mobile environment is a challenge for further research and development.

Advantages

1) Experimental results on dataset showed that the solution is very efficient as it has 99.39% true positive rate and only 1.25 % false positive rate.
2) The proposed approach also has good accuracy as compared to other existing anti-phishing solutions.

Drawbacks

1) The feature set of this phishing detection approach entirely depends on the URL and source code of the web-site, which can detect the webpages written in HTML code only.

4. The detection of sensitive identity theft websites using a working framework based on Routhu Srinivasa Rao machine learning, Alwyn Roshan Pais. This paper, suggests a novel model for the detection of criminal sites for the theft of sensitive information using the colour features of websites. The method described on the paper identifies criminal sites for stealing sensitive information based on URLs, Website content and third-party services using machine learning algorithms. The model shown also detects cybercrime sites that imitate legitimate sites by retrieving website content with an image, most of which are anti- phishing scams. This model finds zero-day phishing sites and list-based strategies fail to find. With the help of such rich agents of dominating factors, the proposed model is able to achieve a high acquisition rate of 99.55% and a low false rate of 0.45% using the oblique Random algorithm.

Advantages

1) The model detects zero-day phishing sites as well which the list-based techniques fail to detect.

2) With the help of such rich set of heuristic features, the model proposed is able to achieve high detection rate of 99.55% and low false positive rate of 0.45% using oblique Random Forest algorithm.

Drawbacks

1) The model purely depends on the quality and quantity of the training set. Hence, training the model with a small dataset and having more duplicate entries in it may not give effective results in phishing detection.
2) model uses third-party services for classification of websites resulting in dependency on the speed of third- party services.
3) model does not work on the websites which use captcha checking before loading the websites because Jsoup connects to the URL to get the source code directly, does not involve handling of captcha verification

5. Jun Ho Huh and Hyoungshick Kin has a phishing detection heuristic-based model in their paper "Phishing Detection with Popular Search Engines: Simple and Effective". In the proposed work, the authors used a full URL of a website which a user intends to access as a search string and on the basis of the results returned, a ranking was provided and the websites were classified accordingly. The model had ranked the legitimate websites at the top whereas majority of the phishing websites were not ranked at all. The websites were classified using 4 main machine learning algorithms namely Linear Discriminant Analysis, Naïve Bayesian, K- Nearest Neighbour and Support Vector Machine. The KNN algorithm performed the best with a true positive rate of 98% and false negative rate of 2%. The most challenging task was the extraction of right features that correctly identifies phishing websites. If the task is not managed properly, then the results are bound to encounter false positives and false negatives.

Advantages

1) It is effective against the new websites and cache hundred's and thounsand's of new and old web pages.
2) It is an easy to implement project and can be deployed

Drawbacks

1) Need to keep track of the websites updates continuously else the model may not produce accurate results.
2) Usage of a small sample pool.

6. Nuttapong Sangalerdsinlapachai and Arnon Rungsawang have proposed a method to phishing web pages using Classifier Ensemble method. Depending on the features returned from Carnegie Mellon Anti phishing and Network Analysis Tool (CANTINA), the authors have added, modified features to train a machine learning model. In this study, six machine learning techniques had been used for improving blocking efficiency. These techniques included AdaBoost, J 48 Decision Tree, Naïve Bayes, Neural Network, Random Forest and Support Vector machine. Additional features such as home page similarity features were added. The model was trained on 500 phishing web pages and 500 non-phishing web pages. The proposed method resulted in 30% boost in accuracy from the traditional heuristic method. The main challenge in this work was maintenance if a proper dataset.

Advantages

1) The features added managed to boost accuracy up to 15 and 20 %.
2) The proposed method is very easy to implement.

Drawbacks

1) The developed features have to be tuned up to work better
2) Lack of proper dataset collection.

7. Anandita has proposed a "Novel ensemble technique to detect phishing emails". In the present work, Ensemble learning approach has been used for phishing email detection. The model comprises of three phases- pre-processing, feature examining, classification stage. The emails have been classified as phish or ham using the prediction of Ensemble Classifier of the five ML Algorithms-Gaussian Naive Bayes, Bernoulli Naïve Bayes, Random Forest Classifier, K-Nearest Neighbours, Support Vector Machines. Clearly, the accuracy has improved from 94.0984% (which is the maximum attained accuracy value by Random

Forest Classifier) to 98.02% (by Ensemble learning). The proposed methodology is rooted upon a set of 15 features obtained by processing of header and body i.e. text part and HTML part of each email in the email data set comprising of 2550 ham emails and 500 phish emails. Random Forest Classifier and Bernoulli Naïve Bayes produced best results among the 5 compared algorithms. The challenging phase of this paper was ensuring the emails were pre-processed for accurate results.

Advantage

1) Enhancement In the accuracy levels for the machine learning algorithm.

Drawbacks

1) Haven't utilized advanced algorithms making it difficult to guarantee novelty of this model.

8. Jian Mao has proposed a solution in his paper work "Phishing-Alarm: Robust and Efficient Phishing Detection via Page Component Similarity". This method developed is formally called the 'Phishing-Alarm" which detects phishing attacks using features that are hard to evade by the attackers. In particular an algorithm was made to quantify the suspiciousness ratings of the Webpages based on similarity of visual appearance between web pages. Visual similarity has been defined using CSS. Ratings of the web pages were made on weighted page-component similarity. The final results obtained from different approaches namely CANTINA, Corbetta, Belabed , Zhang et.al , CASTLE were compared with the Phising-Alarm technique developed and it showed a remarkable precision of 100% with a recall of 97.2% and F1 score of 99%. This method managed to stand out from the other methods and has been added as an extension to Google Chrome browser. The crucial part of this paper work was being able to classify very similar webpage which made it mandatory to develop a new algorithm.

Advantage

1) Usage of advanced technique making it difficult to be evaded by attackers.
2) The method was tested on a large number of datasets.
3) Detection performance is independent from the language of the pages.

Drawbacks

1) The main limitations with using blacklists and whitelists is that they can only classify previously-known phishing or legitimate websites. Inevitably, these lists are not very effective when it comes to identifying a newly formed website.

9. Samuel Marchal has proposed an automated Phishing detection system in his paper named "Phish Storm: Detecting Phishing with Streaming Analytics". The proposed system has the potential to analyse any URL and predict if the site is a potential phishing website. Phish Storm has the ability to interface with any email server or HTTP proxy. A new concept called intra-URL relatedness has been modelled to evaluate and extract features from words and compose a URL based query data from Google and Yahoo search engines. These featured are later used in machine learning based classification to detect phishing URL's. The technique developed was tested on 96018 phishing and legitimate URL's that resulted in a correct classification rate of 94.91% with 1.44% false positives. Creating a dedicated

corpus to be used with existing methods would be helpful but challenging, word relatedness can be dynamically inferred from search engine query data.

Advantage

1) This experiment yielded a classification accuracy of 94.91% with a low false positive rate of 1.44%.
2) Dynamically computed the risk score.

Drawbacks

1) Minimizing the number of legitimate websites classified as phishing reduced the accuracy levels.
2) Testing on multiple ranges to finally find the appropriate soft prediction range.

10. Framework for a novel that uses the Bayesian approach to the detection of sensitive content theft presented by Haijun Zhang in his paper entitled "Visual Text Against Visual Fraud: A Bayesian Approach". The model focuses on text and visual content to measure the similarity between a protected web page and suspicious web pages. A text is read, a person separates the image, and an algorithm that uses student-level results in this paper. A prominent feature of this paper is to examine the Bayesian model to measure the same limit. This is necessary for the classifier to determine the category of the web page and to determine whether the web page is attractive or not. In the text classifier, the naive Bayes rule is used to calculate the probability of a web page being diverted. Image classification, the earth mover distance works to measure visual similarity, and the Bayesian model is designed to determine the limit. In the data compilation algorithm, Bayes view is used to combine the results of the division of written and visual content. The effectiveness of the proposed method was tested on large data collected in cases of actual theft. Test results have shown that the text classifier and the graphic design category bring promising results, the best fusion algorithm for any of the segregated ones, and our model can be adapted to different criminal cases.

    Advantage

    1) The new features of this framework can be represented by a text classifier, an image classifier, and a fusion algorithm.
    2) The content-based model can be easily embedded into current industrial anti-phishing systems.

    Drawbacks

    1) Haven’t solved solve the knowledge updating problem in current probabilistic model

11. "In-depth learning strategies for the detection of cybercrime websites" by M. SOMESHA, ALWYN ROSHAN PAIS, ROUTHU SRINIVASA THEM and VIKRAM SINGH RATHOUR. This paper suggested an in-depth reading model to determine the authenticity of a given website. Heuristic and third-party services-based URLs used to train in-depth learning models. the number of features was reduced and reduced dependence on third-party services led to a significant recognition gain of 99.57%. The features were also tested with a variety of in-depth models based on CNN, DNN and LSTM, and we found 99.57% accuracy with LSTM, 99.43% with CNN and 99.52% with DNN. LSTM and DNN worked with better 10-factor results than the previous 18-factor machine learning function.

    Advantages

    1) It performs better with 10 features, and achieves training accuracy of 99.29% and testing accuracy of 99.43%.

2) The proposed model with LSTM outperforms other proposed models with an accuracy of 99.57%, which is an improvement over the previous work (99.5%) with minimal features.

Drawbacks

1) The proposed model might fail to detect phishing sites that use embedded objects such as flash, java scripts and HTML files to replace textual content.
2) The proposed model is dependent on third-party services, the non-availability of these services will limit the performance of our work.

12. "A Feature-Based Feature Based on the Swarm Foundation for Improving Access to Intelligent Phishing Website" by WALEED ALI AND SHARAF MALEBARY. In this study, a mechanism has been suggested for the detection of explicit crime of sensitive information according to the PSO. In the PSO-based feature test, the website features were weighted with the appropriate weight using PSO to improve the detection of criminal websites for stealing sensitive information. The test results showed that the accuracy of the BPNN, SVM, NB, C4.5, RF, and KNN divisions was significantly increased after applying the proposed PSO weight. In addition, BPNN, SVM, C4.5, RF, and kNN developed a PSO-based feature to better TPR, TNR, FPR, and FNR compared to single-machine learning models. This has shown that advanced machine learning models with a PSO-based weight are able to successfully detect and distinguish both fraudulent and official websites, respectively.

Advantages

1) The machine learning models improved with the proposed PSO-based feature weighting were able to successfully detect and classify both phishing and legitimate websites, respectively

Drawbacks

1) The proposed PSO-based feature weighting, used the original PSO, which utilized the classification accuracy as a fitness objective function during the process of feature weighting. So, the time of feature evaluation and weighting may require a longer time based on the nature of the machine learning algorithm used.
2) Other phishing websites datasets should be used to validate and evaluate the proposed PSO-based feature weighting.

13. “Detection of phishing websites using a novel two-fold ensemble model” Kalyan Nagaraj, Biplab Bhattacharjee, Amulyashree Sridhar and Sharvani GS. The findings from this study are radiated in three phases: the first phase focuses on identifying significant attributes which leads to phishing attacks. The next phase reflects the importance and evaluation of different variants of individual and ensemble machine learners toward detection of phishing websites. The last phase proposes an intelligent detection system supported by RF_NN model for predicting the probability of phishing websites. The initial phase of this study highlights the findings from feature selection algorithm Boruta, which identifies six relevant features among 30 parameters which played a role in phishing websites. These six parameters predict the occurrence of phishing websites accurately for 93.41 per cent of the data instances. Among the four individual classifiers, FF_ANN variant shows promising results with the highest accuracy of 72.45 per cent. Twofold ensemble models are created using random forest and feedforward neural network (RF_NN), bagging and feedforward neural network (Bagging_NN) along with boosting and feedforward neural network (Boosting_NN). It was noted that the performance of RF_NN model increases drastically.

Advantages

1) Among the four individual classifiers, FF_ANN variant shows promising results with the highest accuracy of 72.45 per cent.
2) Twofold ensemble models are created using random forest and feedforward neural network (RF_NN), bagging and feedforward neural network (Bagging_NN) along with boosting and feedforward neural network (Boosting_NN). It was noted that the performance of RF_NN model increases drastically

Drawbacks

1) The research data set used in this study is publicly available and easy to analyse. Comparative analysis with other real-time data sets of recent origin must be performed to ensure generalization of the model against various security breaches.
2) Different variants of phishing threats must be detected rather than focusing particularly toward phishing website detection.

14. R.S Rao et al proposed a novel classification approach that uses heuristic based feature extraction approach. In this, they have classified extracted features into three categories such as URL Obfuscation features, Third-Party-based features, Hyperlink-based features. Moreover, the proposed technique gives 99.55% accuracy. Drawback of this is that as this model uses third party features, classification of websites dependent on speed of third-party services. Also, this model purely depends on the quality and quantity of the training set and Broken links feature extraction has a limitation of more execution time for the websites with larger number of links.

15. Chunlin proposed an approach that primarily focuses on character frequency features. In this they have combined statistical analysis of URL with machine learning technique to get results that are more accurate for classification of malicious URLs. Also they have compared six machine-learning algorithms to verify the effectiveness of the proposed algorithm which gives 99.7% precision with false positive rate less than 0.4%.

16. Sudhanshu used an association data mining approach. They have proposed rule-based classification techniques for phishing website detection. They have concluded that association classification algorithms are better than any other algorithms because of their simple rule transformation. They achieved 92.67% accuracy by extracting 16 features but this is not up to mark so the proposed algorithm can be enhanced for efficient detection rate.

17. M. Amaad has introduced a pure model for the segmentation of criminal websites to steal sensitive information. In this paper, the proposed model is developed in two phases. In Section 1, each makes a separation strategy, and selects three excellent models based on high accuracy and other methods of operation. While in the second phase, they re-assembled each model with each of the three best models and created a hybrid model that offers better accuracy than the individual model. They found 97.75% accuracy in test data. There is a limit to this model that it takes more time to build a hybrid model.

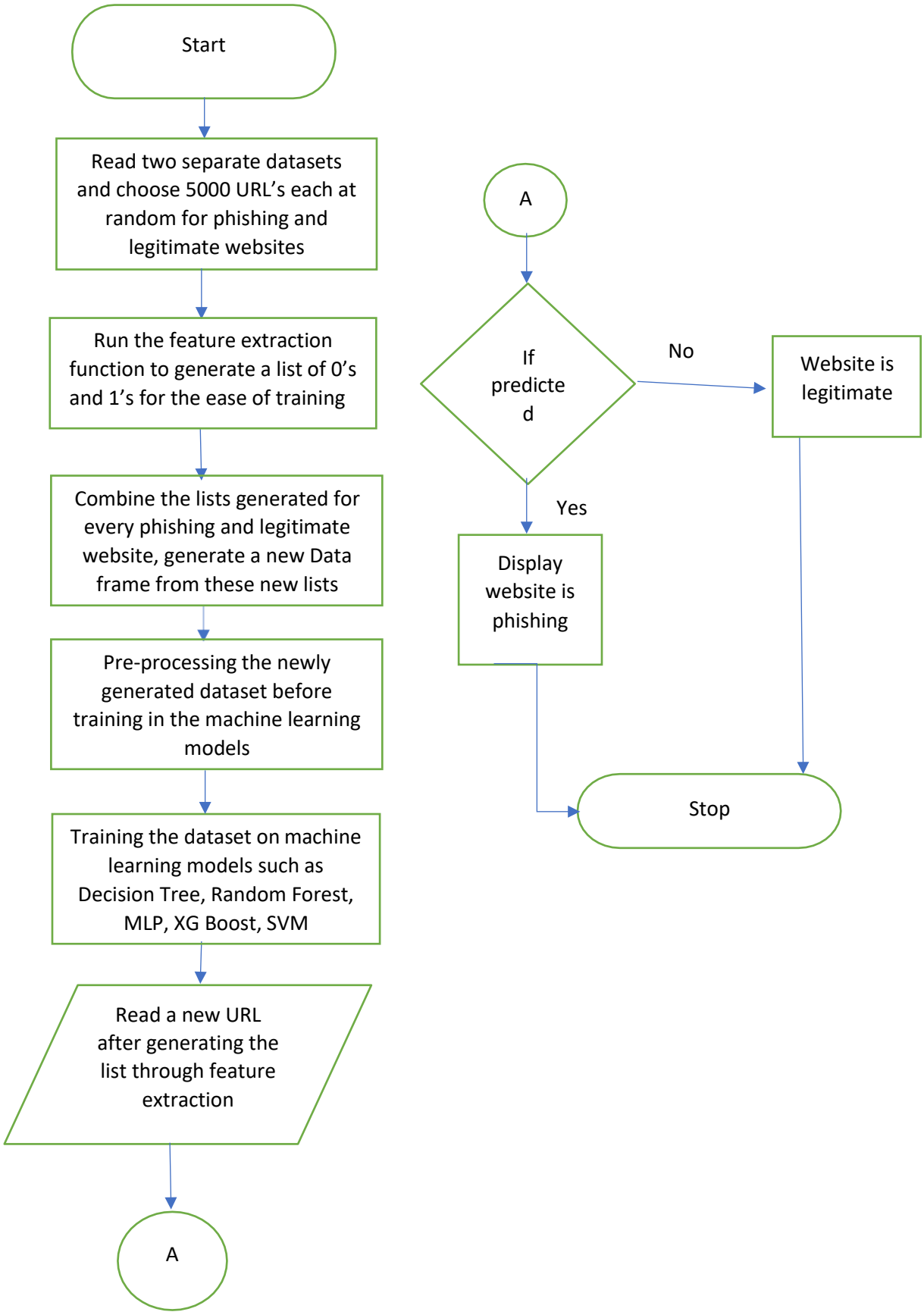
18. Hossein et al. has developed an open source framework known as “New-Phish”. For sensitive identity theft websites, machine learning data can be created using this framework. In this case, they have used the reduced features set and used the python of building materials such as lexical features, URL-supported feature, network-based features and a domain-based feature.

19. Ankit Kumar Jain has proposed a method of cheating a novel that resists removing features from the customer side only. The proposed method is fast and reliable as it does not depend on the third party. but only removes features from URL and source code. In this paper, they found 99.09% of the total accuracy of the website. This paper concludes that this method

has limitations as it can see a web page written in HTML. A non-HTML web page cannot be accessed this way.

20. Priyanka proposed a novel approach by combining two or more algorithms. In this paper, the author has implemented two algorithms Adaline and Backpropion along with SVM for getting a good detection rate and classification purpose. Pradeepthi et al. In this paper, Author studied different classification algorithms and concluded that tree-based classifiers are best and gives better accuracy for phishing URL detection.

FLOW CHART


Start
Read two separate datasets and choose 5000 URL's each at random for phishing and legitimate websites
Run the feature extraction function to generate a list of 0's and 1's for the ease of training
Combine the lists generated for every phishing and legitimate website, generate a new Data frame from these new lists
Pre-processing the newly generated dataset before training in the machine learning models
Training the dataset on machine learning models such as Decision Tree, Random Forest, MLP, XG Boost, SVM
Read a new URL after generating the list through feature extraction
A
A
If predicte d
No
Website is legitimate
Yes
Display website is phishing
Stop

PROPOSED ALGORITHM

1) Collect dataset containing phishing and legitimate websites from the open source platforms.
2) Extract the required features from the URL database.
3) Analyse and pre-process the dataset by using EDA techniques.
4) Divide the dataset into training and testing sets.
5) Run selected machine learning and deep neural network algorithms like SVM, Random Forest, Autoencoder on the dataset.
6) Write a code for displaying the evaluation result considering accuracy metrics.
7) Compare the obtained results for trained models and specify which is better.

METHODOGY USED

DATA COLLECTION: Legitimate URLs are collected from the dataset provided from an open source platform. Phishing URLs are collected from opensource service called Phish Tank. This service provides a set of phishing URLs in multiple formats like csv, json etc. that gets updated hourly.

FEATURE SELECTION: After the Data collection feature selection process has to be done The, following category of features are selected:

1) Address Bar based Features
   - Domain of URL
   - Redirection ‘//’ in URL
   - IP Address in URL
   - ‘http/https’ in Domain name
   - ‘@’ Symbol in URL
   - Using URL Shortening Service
   - Length of URL
   - Prefix or Suffix "-" in Domain
   - Depth of URL

2) Domain based Features
   - DNS Record
   - Age of Domain
   - Website Traffic
   - End Period of Domain

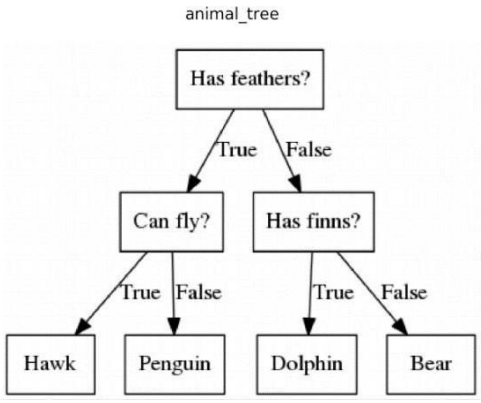
3) HTML & Java-script based Feature
   - Iframe Redirection
   - Disabling Right Click
   - Status Bar Customization
   - Website Forwarding

MACHINE LEARNING MODEL USED

This data set comes under classification problem, as the input URL is classified as phishing (1) or legitimate (0). The machine learning models (classification) considered to train the dataset are:

1) Decision Tree

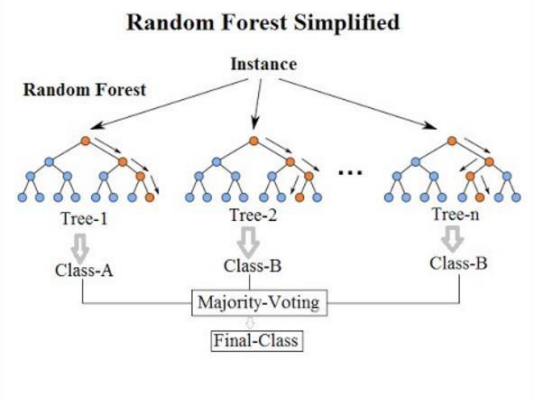
Decision tree learning is one of the predictive modelling approaches used in statistics, data mining and **machine learning**. It uses a **decision tree** (as a predictive model) to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves).



2) Random Forest

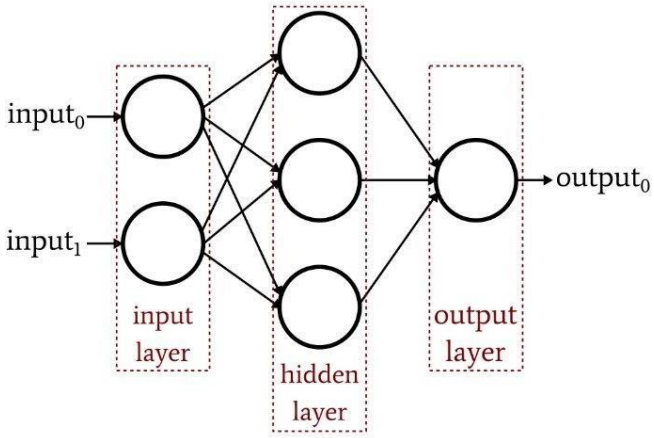
Random forest is a supervised learning algorithm which is used for both classification as well as regression. But however, it is mainly used for classification problems. As we know that a forest is made up of trees and more trees means more robust forest. Similarly, random forest algorithm creates decision trees on data samples and then gets the prediction from each of them

and finally selects the best solution by means of voting. It is an ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result.



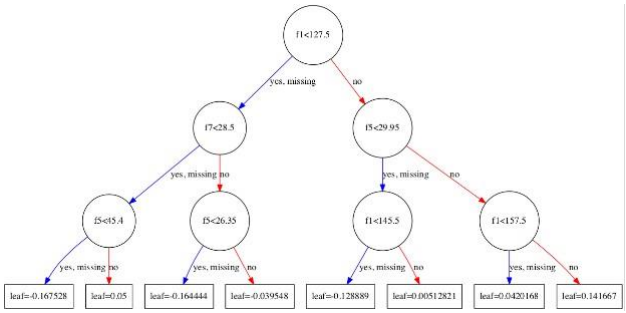
3) Multilayer Perceptron

A multilayer perceptron (MLP) is a class of feedforward artificial neural network (ANN). The term MLP is used ambiguously, sometimes loosely to any feedforward ANN, sometimes strictly to refer to networks composed of multiple layers of perceptrons (with threshold activation). Multilayer perceptrons are sometimes colloquially referred to as "vanilla" neural networks, especially when they have a single hidden layer.



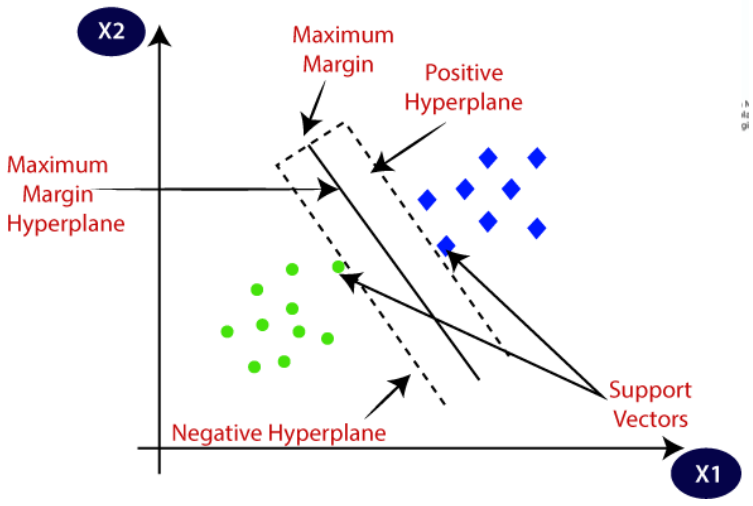
4) XG Boost

XG Boost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. In prediction problems involving unstructured data (images, text, etc.) artificial neural networks tend to outperform all other algorithms or frameworks.



5) Support Vector Machines

More formally, a support-vector machine constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks like outliers detection. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin, the lower the generalization error of the classifier.



MODEL EVALUATION

The models are evaluated, and the considered metric is accuracy. All the above ML algorithm accuracy on the data set will be calculated and the highest will be stored and will selected for the further deployment of that model.

The accuracy can be defined as the percentage of correctly classified instances

(TP + TN)/ (TP + TN + FP + FN). where TP, FN, FP and TN represent the number of true positives, false negatives, false positives and true negatives, respectively.

The true positive (TP) and true negatives (TN) are correct classification. A false positive (FP) is when the outcome of the algorithm is incorrectly predicted, when the in reality it is actually present in the image.

CODE SNIPPITS

Phishing Website URL's dataset

```
#loading the phishing URLs data to dataframe
data0 = pd.read_csv(r'C:\Users\DELL\Documents\Info_sec_proj\Phishing-Website-Detection\DataFiles\2.online-valid.csv')
data0.head()
```

	phish_id	url	phish_detail_url	submission_time	verified	verification_time	online	targ
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	Oth
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	Oth
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	Facebo
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	Facebo
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	Micros

5000 URL's sampled at random from the phishing website dataset

```
#Collecting 5,000 Phishing URLs randomly
phishurl = data0.sample(n = 5000, random_state = 12).copy()
phishurl = phishurl.reset_index(drop=True)
phishurl.head()
```

	phish_id	url	phish_detail_url	submission_time	verified
0	6514946	http://confirmprofileaccount.com/	http://www.phishtank.com/phish_detail.php?phis...	2020-04-19T11:06:55+00:00	yes
1	4927651	http://www.marreme.com/MasterAdmin/04mop.html	http://www.phishtank.com/phish_detail.php?phis...	2017-04-04T19:35:54+00:00	yes
2	5116976	http://modsecpaststudents.com/review/	http://www.phishtank.com/phish_detail.php?phis...	2017-07-25T18:48:30+00:00	yes
3	6356131	https://docs.google.com/forms/d/e/1FAIpQLScL6L...	http://www.phishtank.com/phish_detail.php?phis...	2020-01-13T20:13:37+00:00	yes
4	6535965	https://oportunidadedasemana.com/americanas//?...	http://www.phishtank.com/phish_detail.php?phis...	2020-04-29T00:01:03+00:00	yes

Legitimate Website URL's dataset

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

```
#Loading legitimate files
data1 = pd.read_csv(r'C:\Users\DELL\Documents\Info_sec_proj\Phishing-Website-Detection\DataFiles\1.Beni
data1.columns = ['URLs']
data1.head()
```

	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...

5000 URL's sampled at random from the legitimate website dataset

```
#Collecting 5,000 Legitimate URLs randomly
legiurl = data1.sample(n = 5000, random_state = 12).copy()
legiurl = legiurl.reset_index(drop=True)
legiurl.head()
```

	URLs
0	http://graphicriver.net/search?date=this-month...
1	http://ecnavi.jp/redirect/?url=http://www.cros...
2	https://hubpages.com/signin?explain=follow+Hub...
3	http://extratorrent.cc/torrent/4190536/AOMEI+B...
4	http://icicibank.com/Personal-Banking/offers/o...

Feature Extraction

1) Address Based Features

1.1) Domain of URL

```
# 1.Domain of the URL (Domain)
def getDomain(url):
    domain = urlparse(url).netloc
    if re.match(r"^www.",domain):
        domain = domain.replace("www.","")
    return domain
```

```
phishurl.iloc[0]['url']
```

```
'http://confirmprofileaccount.com/'
```

```
domain=urlparse(phishurl.iloc[0]['url'])
```

```
domain=domain.netloc
```

```
if re.match(r"^www.",domain):
        domain = domain.replace("www.","")
```

```
domain
```

```
'confirmprofileaccount.com'
```

Figure 1 Extracting the domain part in the URL

1.2) IP Address in the URL

If IP address is contained in the URL, site is phishing (return 1), else site is legitimate (return 0)

```
# 2.Checks for IP address in URL (Have_IP)
def havingIP(url):
    try:
        domain=(urlparse(url).netloc).strip()
        ipaddress.ip_address(domain)
        ip = 1
    except:
        ip = 0
    return ip
```

```
url=phishurl.iloc[0]['url']
```

```
domain=(urlparse(url).netloc).strip()
```

```
try:
    domain=(urlparse(url).netloc).strip()
    ipaddress.ip_address(domain)
    ip = 1
except:
    ip = 0
```

```
ip
```

```
0
```

Figure 2 Checking for the presence of IP address in the URL

1.3) "@" symbol in URL

If @ symbol is contained in the URL, site is phishing (return 1), else site is legitimate (return 0)

```
# 3.Checks the presence of @ in URL (Have_At)
def haveAtSign(url):
    if "@" in url:
        at = 1

    else:
        at = 0
    return at
```

Figure 3 Checking for the presence of @ symbol in the URL

1.4) Length of the URL

If length of the URL>=54, site is phishing (return 1), else site is legitimate (return 0)

```
# 4.Finding the length of URL and categorizing (URL_Length)
def getLength(url):
    if len(url) < 54:
        length = 0
    else:
        length = 1
    return length
```

```
len(phishurl.iloc[20]['url'])
```

```
101
```

Figure 4 Checking for the length of the URL

1.5) Depth of the URL

Calculates the number of subpages in the URL by calculating the count of forward slashes

```
# 5.Gives number of '/' in URL (URL_Depth)
def getDepth(url):
    s = urlparse(url).path.split('/')
    depth = 0
    for j in range(len(s)):
        if len(s[j]) != 0:
            depth = depth+1
    return depth
```

Figure 5 Checking for the depth of the URL

1.6) Redirection “//” in URL

If the "//" is anywhere in the URL apart from after the protocol, thee value assigned to this feature is 1 (phishing) or else 0 (legitimate).

```
# 6.Checking for redirection '//' in the url (Redirection)
def redirection(url):
    pos = url.rfind('//')
    if pos > 5:
        if pos > 6:
            return 1
        else:
            return 0
    else:
        return 0
```

Figure 6 Checking for the position of // in URL

1.7) "http/https" in Domain name

If the URL has "http/https" in the domain part, the value assigned to this feature is 1 (phishing) or else 0 (legitimate).

```
# 7.Existence of “HTTPS” Token in the Domain Part of the URL (https_Domain)
def httpDomain(url):
    domain = urlparse(url).netloc
    if 'https' in domain:
        return 1
    else:
        return 0
```

Figure 7 Checking for the presence of http/https in the URL

1.8) Using URL Shortening Services “Tiny URL”

If the URL is using Shortening Services, the value assigned to this feature is 1 (phishing) or else 0 (legitimate).

```
# 8. Checking for Shortening Services in URL (Tiny_URL)
def tinyURL(url):
    match=re.search(shortening_services,url)
    if match:
        return 1
    else:
        return 0
```

Figure 8 Checking for the presence of shortening services in the URL

1.9) Prefix or Suffix "-" in Domain

If the URL has '-' symbol in the domain part of the URL, the value assigned to this feature is 1 (phishing) or else 0 (legitimate).

```
# 9.Checking for Prefix or Suffix Separated by (-) in the Domain (Prefix/Su
def prefixSuffix(url):
    if '-' in urlparse(url).netloc:
        return 1            # phishing
    else:
        return 0            # legitimate
```

Figure 9 Checking for the presence of '-' symbol in the URL

2) Domain Based Features

2.1) DNS Record

For phishing websites, either the claimed identity is not recognized by the WHOIS database or no records founded for the hostname. If the DNS record is empty or not found then, the value assigned to this feature is 1 (phishing) or else 0 (legitimate).

2.2) Web Traffic

If the rank of the domain > 100000, the value of this feature is 1 (phishing) else 0 (legitimate).

```
# 12.Web traffic (Web_Traffic)
def web_traffic(url):
    try:
        #Filling the whitespaces in the URL if any
        url = urllib.parse.quote(url)
        rank = int(BeautifulSoup(urllib.request.urlopen("http://data.alexa.com/data?cli=10&dat=
    except TypeError:
        return 1
    if rank >100000:
        return 1
    else:
        return 0
```

Figure 10 Returning the rank of URL

2.3) Age od Domain

If age of domain < 12 months, the value of this feature is 1 (phishing) else 0 (legitimate).

```
# 13.Survival time of domain: The difference between termination time and creation time (Domain
def domainAge(domain_name):
    creation_date = domain_name.creation_date
    expiration_date = domain_name.expiration_date
    if (isinstance(creation_date,str) or isinstance(expiration_date,str)):
        try:
            creation_date = datetime.strptime(creation_date,'%Y-%m-%d')
            expiration_date = datetime.strptime(expiration_date,"%Y-%m-%d")
        except:
            return 1
    if ((expiration_date is None) or (creation_date is None)):
        return 1
    elif ((type(expiration_date) is list) or (type(creation_date) is list)):
        return 0
    else:
        ageofdomain = abs((expiration_date - creation_date).days)
        if ((ageofdomain/30) < 6):
            age = 1
        else:
            age = 0

    return age
```

Figure 11 Checking for the age of the domain of URL

2.4) End Period of Domain

If end period of domain < 6 months, the value of this feature is 1 (phishing) else 0 (legitimate).

```
# 14.End time of domain: The difference between termination time and current time (Domain_End)
def domainEnd(domain_name):
    expiration_date = domain_name.expiration_date
    if isinstance(expiration_date,str):
        try:
            expiration_date = datetime.strptime(expiration_date,"%Y-%m-%d")
        except:
            return 1
    if (expiration_date is None):
        return 1
    elif (type(expiration_date) is list):
        return 0

    else:
        today = datetime.now()
        end = abs((expiration_date - today).days)
        if ((end/30) < 6):
            end = 0
        else:
            end = 1
    return end
```

Figure 12 Checking for End period of Domain

3) HTML and JavaScript based Features

3.1) IFrame Redirection

If the iframe is empty or response is not found then, the value assigned to this feature is 1 (phishing) or else 0(legitimate).

```
# 15. IFrame Redirection (iFrame)
def iframe(response):
    if response == "":
        return 1

    else:
        if re.match(r"[<iframe>|<frameBorder>]", response.text): #findall
            return 0
        else:
            return 1
```

Figure 13 Checking for the presence of Iframe in the URL

3.2) Status Bar Customization

If the response is empty or on mouseover is found then, the value assigned to this feature is 1 (phishing) or else 0(legitimate).

```
# 16.Checks the effect of mouse over on status bar (Mouse_Over)
def mouseOver(response):
    if response == "" :
        return 1
    else:
        if re.findall("<script>.+onmouseover.+</script>", response.text):
            return 1
        else:
            return 0
```

Figure 14 Checking for mouseover

3.3) Disabling Right Click

If the response is empty or on mouseover is not found then, the value assigned to this feature is 1 (phishing) or else 0(legitimate).

```
# 17.Checks the status of the right click attribute (Right_Click)
def rightClick(response):
    if response == "":
        return 1
    else:
        if re.findall(r"event.button ?== ?2", response.text):
            return 0
        else:
            return 1
```

Figure 15 Disabling the right click

3.4) Website Forwarding

We find that legitimate websites have been redirected one time max. On the other hand, phishing websites containing this feature have been redirected at least 4 times.

```
# 18.Checks the number of forwardings (Web_Forwards)
def forwarding(response):
    if response == "":
        return 1
    else:
        if len(response.history) <= 2:
            return 0
        else:
            return 1
```

Figure 16 Checking for the number of redirections

Machine Learning Models & Training

This data set comes under classification problem, as the input URL is classified as phishing (1) or legitimate (0). The supervised machine learning models (classification) considered to train the dataset in this notebook are:

1) Decision Tree
2) Random Forest
3) Multilayer Perceptrons
4) XGBoost
5) Support Vector Machines

1) Decision Tree Classifier

```
# 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)
```

2) Random Forest

```
# 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)
```

3) Multilayer Perceptrons

```
# Multilayer Perceptrons model
from sklearn.neural_network import MLPClassifier

# instantiate the model
mlp = MLPClassifier(alpha=0.001, hidden_layer_sizes=([100,100,100]))

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

4) XG Boost

```
#XGBoost Classification model
from xgboost import XGBClassifier

# instantiate the model
xgb = XGBClassifier(learning_rate=0.4,max_depth=7)
#fit the model
xgb.fit(X_train, y_train)
```

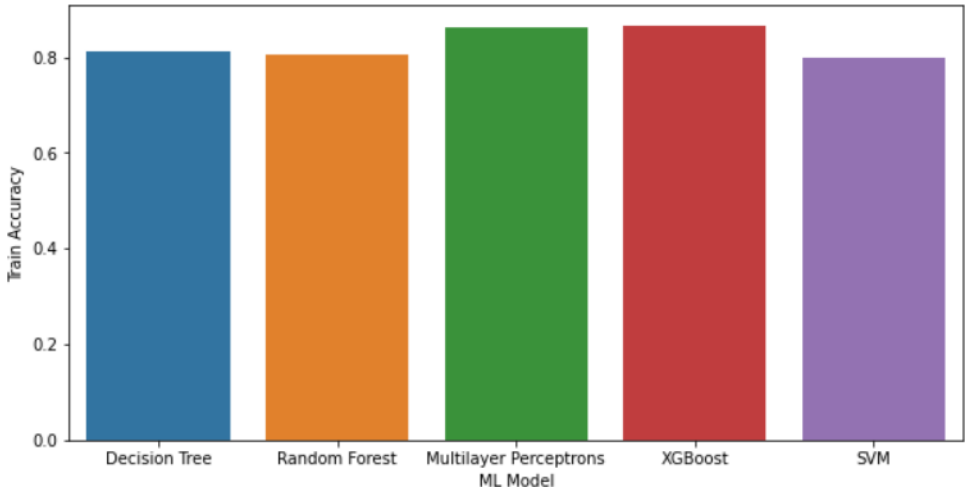
5) Support Vector Machines

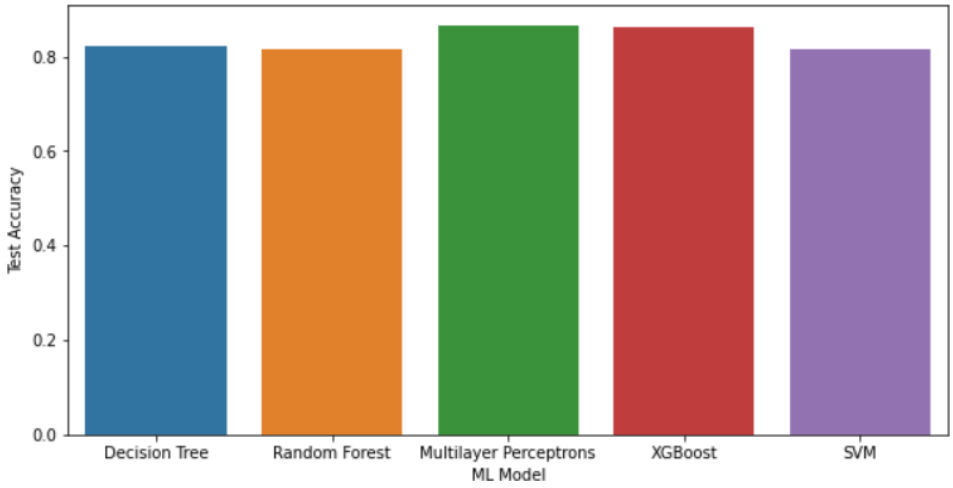
```
#Support vector machine model
from sklearn.svm import SVC

# instantiate the model
svm = SVC(kernel='linear', C=1.0, random_state=12)
#fit the model
svm.fit(X_train, y_train)
```

Comparison of Models

The models were compared. For the above comparison, it is clear that the XGBoost Classifier works well with this dataset.

	ML Model	Train Accuracy	Test Accuracy
0	Decision Tree	0.811	0.822
1	Random Forest	0.804	0.815
2	Multilayer Perceptrons	0.864	0.866
3	XGBoost	0.866	0.864
4	SVM	0.799	0.814





RESULTS OBTAINED

1) Sample Testing

We test the model with a known phishing website to check if the model works.

```
test_features=[['confirmprofileaccount.com',
  0,
  0,
  0,
  0,
  0,
  0,
  1,
  0,
  0,
  1,
  0,
  0,
  1,
  1,
  1,
  1,
  None]]
```

```
al=[]
al=svm.predict(X_sample)
if(al[0]==1 and test_features[0][0] not in l ):
  print("It's a phished website")
else:
  print("It's not a phished website")
```

```
#converting the list to dataframe
feature_names = ['Domain', 'Have_IP', 'Have_At', 'URL_Length', 'URL_Depth','Redirection',
                     'https_Domain', 'TinyURL', 'Prefix/Suffix', 'DNS_Record', 'Web_Traffic',
                     'Domain_Age', 'Domain_End', 'iFrame', 'Mouse_Over','Right_Click', 'Web_Fo

testing = pd.DataFrame(test_features, columns= feature_names)
testing.head()
```

Output

It's a phished website.

REFERENCES

[1] Ozgur Koray Sahingoz , Ebubekir Buber , Onder Demir , Banu Diri ,"Machine learning based phishing detection from URLs " September 2018, ELSEIVER.

[2] Shweta Sankhwar , Dhirendra Pandey and R.A Khan , "Email Phishing: An Enhanced Classification Model to Detect Malicious URLs ", April 2019 , EAI.

[3] Ankit Kumar Jain · B. B. Gupta, "Towards detection of phishing websites on client-side using machine learning based approach", December 2017, Springer.

[4] Routhu Srinivasa Rao and Alwyn Roshan Pais, " Detection of phishing websites using an efficient feature-based machine learning framework ", January 2018, Springer.

[5] Jun Ho Huh and Hyoungshick Kim, "Phishing Detection with Popular Search Engines: Simple and Effective", May 12-13 2011, Springer.

[6] Nuttapong Sanglerdsinlapachai, Arnon Rungsawang, "Web Phishing Detection Using Classifier Ensemble", November 2010, ACM.

[7] Anandita, Dhirendra Pratap Yadav, Priyanka Paliwal, Divya Kumar, Rajesh Tripathi, "A Novel Ensemble Based Identification of Phishing E-Mails", February 2017, 2017.

[8] JIAN MAO, WENQIAN TIAN1, PEI LI, TAO WEI AND ZHENKAI LIANG3, "Phishing-Alarm: Robust and Efficient Phishing Detection via Page Component Similarity", August 23 2017, IEEE transaction.

[9] Samuel Marchal , Jérôme François, Radu State, and Thomas Engel ,"Phish Storm: Detecting Phishing With Streaming Analytics", December 2014 , IEEE transaction

[10] Haijun Zhang, Gang Liu, Tommy W. S. Chow, Senior Member and Wenyin Liu, "Textual and Visual Content-Based Anti-Phishing: A Bayesian Approach", October 2011, IEEE transaction.

[11] M SOMESHA, ALWYN ROSHAN PAIS, ROUTHU SRINIVASA RAO and VIKRAM SINGH RATHOUR, “Efficient deep learning techniques for the detection of phishing websites”, February 2020, Indian academy of Science

[12] WALEED ALI AND SHARAF MALEBARY, ”Particle Swarm Optimization-Based Feature Weighting for Improving Intelligent Phishing Website Detection ”, June 2019, IEEE

[13] Kalyan Nagaraj, Biplab Bhattacharjee, Amulyashree Sridhar and Sharvani GS, “Detection of phishing websites using a novel two fold ensemble model ”, July 2018, Emerald insight.

[14] Routhu Srinivasa Rao1 , Alwyn Roshan Pais ,"Detection of phishing websites using an efficient feature-based machine learning framework", In Springer 2018.

[15] Chunlin Liu, Bo Lang, "Finding effective classifier for malicious URL detection ", 2018, ACM

[16] Sudhanshu Gautam, Kritika Rani and Bansidhar Joshi, "Detecting Phishing Websites Using Rule-Based Classification Algorithm", 2018, Springer.

[17] M. Amaad Ul Haq Tahir, Sohail Asghar, Ayesha Zafar, Saira Gillani , "A Hybrid Model to Detect Phishing-Sites using Supervised Learning Algorithms", In International Conference on Computational Science and Computational Intelligence ,2016,IEEE.

[18] Hossein Shirazi, Kyle Haefner, Indrakshi Ray: Fresh-Phish, "A Framework for Auto-Detection of Phishing Websites", In (International Conference on Information Reuse and Integration (IRI)) ,2017, IEEE.

[19] 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

[20] Priyanka Singh, Yogendra P.S. Maravi, Sanjeev Sharma, “Phishing Websites Detection through Supervised Learning Networks” ,2015, IEEE