# Fraudulent Website Detection Using Machine Learning

Team: Rajesh Chinnaga, Adarsh Reddy, Venkata B

## Abstract

Phishing - Phishing is a type of social engineering where an attacker sends a fraudulent (e.g., spoofed, fake, or otherwise deceptive) message designed to trick a person into revealing sensitive information to the attacker or to deploy malicious software on the victim's infrastructure like ransomware [1]. The goal of the project is to train Machine learning models and deep neural networks that help a user to predict if a website is a phishing website. To achieve this, we collect data from a data set that contains data about normal website URLs as well as phishing website URLs. In addition, we will also extract the required URL and website content-based features. Nowadays, phishing websites have gotten so advanced, and phishing website hackers can trick normal users into submitting their personal information as well as financial information. Thus, it is a cybersecurity menace that needs to be stopped.

## 1. Introduction

1.1 **Problem Statement:** The goal of the project is to predict best classification model using Machine learning and deep neural networks that help a user to identify if a website is a phishing website. The goal is to seek more information about how characteristics like websites with email in the URL, websites with a valid TSL or SSL certificate, websites with IP addresses instead of domain names in the URL, affects the website classification.

1.2 **Importance:** Nowadays, phishing websites have gotten so advanced, and phishing website hackers can trick normal users into submitting their personal information as well as financial information. Thus, it is a cybersecurity menace that needs to be stopped.

1.3 **Literature Survey:** According to Phisher's method of capturing putative users, phishing attempts are divided into different categories. Keyloggers, DNS poisoning, and other attack types include a few of them. Social media platforms that leverage web 2.0 services, such Facebook, and Twitter Short Message services (SMS), file-sharing services for peers, Voice over IP(VoIP) systems where attackers use caller spoofing IDs, and online blogs are some of the initiating procedures used in social engineering. Each type of phishing differs slightly in how the procedure is carried out to deceive the unwary customer. When a hacker sends a potential user an email with a link that takes them to phishing websites, this is known as an email phishing attack.[2]

## 2. Methodology

The goal of the project is to train Machine learning models and deep neural networks that help a user to predict if a website is a phishing website. To achieve this, we collect data from a data set that contains data about normal website URLs as well as phishing website URLs. In addition, we will also extract the required URL and website content-based features.

**Steps:**

- Exploring for datasets that contain both normal website URLs and phishing website URLs.
- The below datasets will be used for detecting the fraud websites:
  1. https://www.unb.ca/cic/datasets/url-2016.html
  2. https://www.phishtank.com/developer_info.php
- Do feature engineering on the dataset and shortlist the features.
- Do data exploration and visualization.
- Split the dataset into training and testing sets.
- The following classification models are used to train the model:
  1. Decision Tree
  2. Random Forest Classifier
  3. Multilayer Perceptron's (MLPs)
  4. XGBoost Classifier
  5. Support Vector Machines
- Comparing the models and evaluating the result
- Conclusion
- FutureWork

## 3. Fraudulent Website Detection Dataset

**Phistank(Phishing URL's) Dataset: [3]**

**phish_id :** The ID number by which Phishtank refers to a phish submission. All data in PhishTank is tied to this ID. This will always be a positive integer.

**phish_detail_url:** PhishTank detail url for the phish, where you can view data about the phish, including a screenshot and the community votes.

**url:** The phish URL. This is always a string, and in the XML feeds may be a CDATA block.

**submission_time:** The date and time at which this phish was reported to Phishtank. This is an ISO 8601 formatted date.

**Verified:** Whether or not this phish has been verified by our community. In these data files, this will always be the string 'yes' since we only supply verified phishes in these files.

**verification_time:** The date and time at which the phish was verified as valid by our community. This is an ISO 8601 formatted date.

**Online:** Whether the phish is online and operational. In these data files, this will always be the string 'yes' since we only supply online phishes in these files.

**Target:** The name of the company or brand the phish is impersonating if it's known.

**UNB(Legit/True URL's) Dataset: [4]**

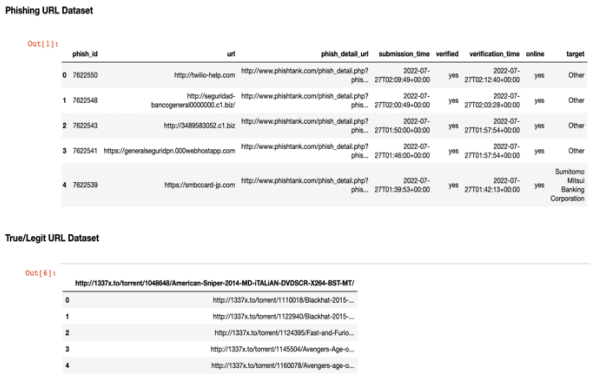
**Benign URLs**: Over 35,300 benign URLs were collected from Alexa top websites. The domains have been passed through a Heritrix web crawler to extract the URLs. Around half a million unique URLs are crawled initially and then passed to remove duplicate and domain only URLs. Later the extracted URLs have been checked through Virustotal to filter the benign URLs.

**Spam URLs**: Around 12,000 spam URLs were collected from the publicly available WEBSPAM-UK2007 dataset.

**Phishing URLs**: Around 10,000 phishing URLs were taken from OpenPhish which is a repository of active phishing sites.

**Malware URLs**: More than 11,500 URLs related to malware websites were obtained from DNS-BH which is a project that maintain list of malware sites.

**Defacement URLs**: More than 45,450 URLs belong to Defacement URL category. They are Alexa ranked trusted websites hosting fraudulent or hidden URL that contains both malicious web pages.


Phishing URL Dataset

Out[1]:

| | phish_id | url | phish_detail_url | submission_time | verified | verification_time | online | target |
|---|---|---|---|---|---|---|---|---|
| 0 | 7622550 | http://twilio-help.com | http://www.phishtank.com/phish_detail.php?phis... | 2022-07-27T02:12:40+00:00 | yes | 2022-07-27T02:12:40+00:00 | yes | Other |
| 1 | 7622548 | http://seguridad-bancogenere80000000.c1.biz/ | http://www.phishtank.com/phish_detail.php?phis... | 2022-07-27T02:00:49+00:00 | yes | 2022-07-27T02:03:28+00:00 | yes | Other |
| 2 | 7622543 | http://3489583052.c1.biz | http://www.phishtank.com/phish_detail.php?phis... | 2022-07-27T01:50:00+00:00 | yes | 2022-07-27T01:57:54+00:00 | yes | Other |
| 3 | 7622541 | https://generalseguridpn.000webhostapp.com | http://www.phishtank.com/phish_detail.php?phis... | 2022-07-27T01:46:00+00:00 | yes | 2022-07-27T01:57:54+00:00 | yes | Other |
| 4 | 7622539 | https://smbccard-jp.com | http://www.phishtank.com/phish_detail.php?phis... | 2022-07-27T01:39:53+00:00 | yes | 2022-07-27T01:42:13+00:00 | yes | Sumitomo Mitsui Banking Corporation |

True/Legit URL Dataset

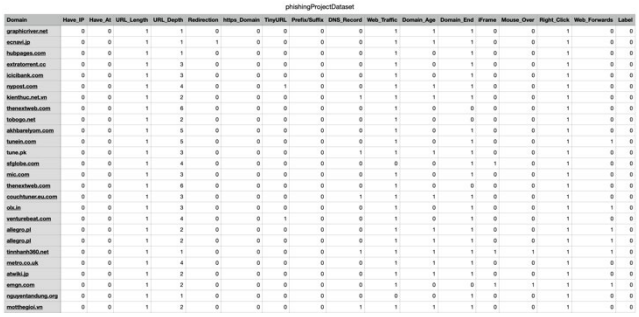
Out[6]:

| | http://1337x.to/torrent/1048648/American-Sniper-2014-MD-ITALIAN-DVDSCR-X264-BST-MT/ |
|---|---|
| 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... |

*Figure: Phishing and True/Legit URL Dataset Information.*

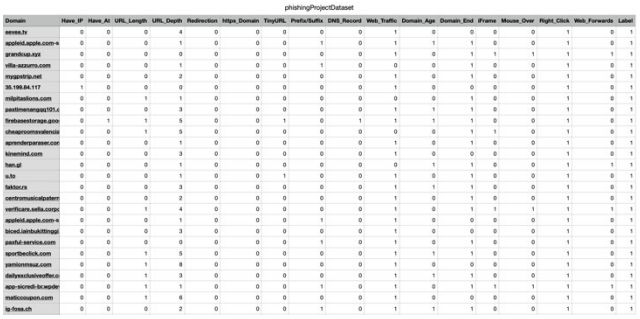
## 4. Data Collection and Feature Extraction

We used Phishtank website and UNB site for getting the phishing and True/Legit datasets. We collected 5000 samples from Phishing dataset and 5000 samples from True/Legit dataset and extracted 15 features out of these URL's and created a final dataset that contains 10,000 samples of Phishing and True/Legit URLs with all the features that we extracted. The attached "IML-Project-DataPreprocessing.ipynb" contains all the code related to data collection and extracting the features.


phishingProjectDataset

| 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_Forwards | Label |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| graphicriver.net | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| ecnavi.jp | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| hulkpages.com | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| extratorrent.cc | 0 | 0 | 1 | 3 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| icicibank.com | 0 | 0 | 1 | 3 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| nypost.com | 0 | 0 | 1 | 4 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| kienthuc.net.vn | 0 | 0 | 1 | 2 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| thenextweb.com | 0 | 0 | 1 | 6 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| tobogo.net | 0 | 0 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| ahlbarstyon.com | 0 | 0 | 1 | 5 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| tunein.com | 0 | 0 | 1 | 5 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| tune.pk | 0 | 0 | 1 | 3 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| sfgiobe.com | 0 | 0 | 1 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| mic.com | 0 | 0 | 1 | 3 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| thenextweb.com | 0 | 0 | 1 | 6 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| couchtuner.eu.com | 0 | 0 | 1 | 3 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| ofo.tv | 0 | 0 | 1 | 3 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| venturebeat.com | 0 | 0 | 1 | 4 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| allegro.pl | 0 | 0 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| allegro.pl | 0 | 0 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| tinnhanh360.net | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| metro.co.uk | 0 | 0 | 1 | 4 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| atwiki.jp | 0 | 0 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| emgn.com | 0 | 0 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 |
| nguyentandung.org | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| motthegioi.vn | 0 | 0 | 1 | 2 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |

*Figure: True/Legit URL's Data with all the extracted Features.*


phishingProjectDataset

| 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_Forwards | Label |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| eenet.tr | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| appleid.apple.com-s | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |
| grandcsxp.xyz | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| villa-azzurro.com | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |
| mypgstrip.net | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 35.198.84.117 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| milpitasilons.com | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |
| pastimenangqq101.c | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |
| firebasestorage.goo | 0 | 1 | 1 | 5 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |
| cheapcomvalensia | 0 | 0 | 1 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| aprenderparaser.cor | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |
| kinemind.com | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| han.gl | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| u.to | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| fakturo.rs | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |
| centromusicalpatern | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |
| verificare.sella.corp | 0 | 0 | 1 | 4 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| appleid.apple.com-s | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| bicad.ianbukittinggi | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| paisful-service.com | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| sportbec6ck.com | 0 | 0 | 1 | 5 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |
| yamionmsuz.com | 0 | 0 | 1 | 8 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| dailyexclusiveoffer.c | 0 | 0 | 1 | 3 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |
| app-sicredi-br.wpdir | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| meticoupon.com | 0 | 0 | 1 | 6 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| ip-fosa.ch | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |

*Figure: Phishing URL's Data with all the extracted Features.*

## 5. Data Exploration

We loaded the final data into phishingProjectDataset.csv which contains all the 10,000 phishing and legit/true URLs and we have created Histograms and correlation Maps to analyze the trends between the data.

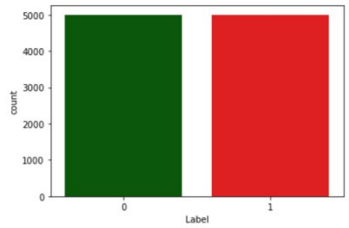



*Figure: Legit and Phishing data with labels, Green which is 0 indicates Legit ULR's data and Red which is 1 indicates Phishing URL's data.*

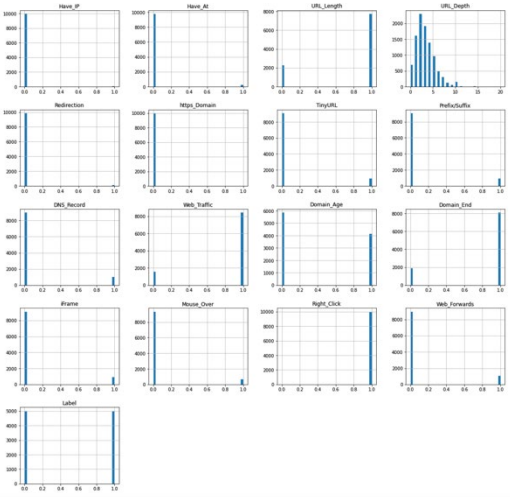



*Figure: Visualization of data using Histogram.*

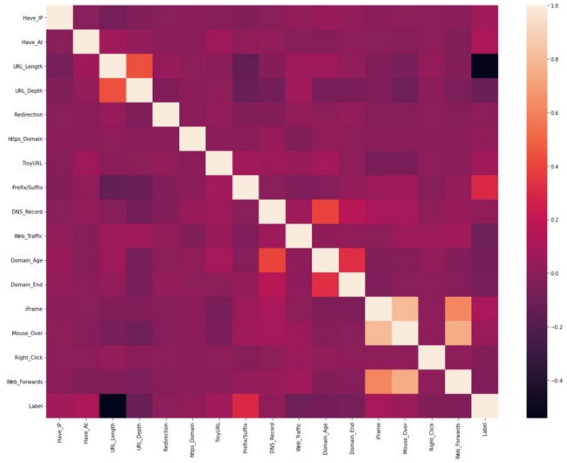


*Figure: Visualization of data using Heatmap.*

## 6. Data Preprocessing

Majority of the features has binary values either 0 or 1, except for the features "Domain" and "URL Depth" which can be seen in the histogram displayed as part of Data Exploration. Domain feature is not required for training the machine learning model or for predicting the target values for the output data set, so we dropped this feature from the features list. Also, URL Depth has the values ranging from 0 to 20 but we felt that this won't be an issue while training the model, so we leave the values at their current levels. Also, we have shuffled the data in order to prevent overfitting during training the model.

| | Have_IP | Have_At | URL_Length | URL_Depth | Redirection | https_Domain | TinyURL | Prefix/Suffix | DNS_Record | \ |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 0 | 1 | 3 | 0 | 0 | 0 | 0 | 0 | |
| **1** | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | |
| **2** | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 1 | |
| **3** | 0 | 0 | 1 | 4 | 0 | 0 | 0 | 0 | 0 | |
| **4** | 0 | 0 | 1 | 4 | 0 | 0 | 0 | 0 | 0 | |

| Web_Traffic | Domain_Age | Domain_End | iFrame | Mouse_Over | Right_Click | Web_Forwards | Label |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |

*Figure: Final Features for training the Machine Learning Models.*

## 7. Training Data using Classification models

We are using the below mentioned classification models by splitting the data into train and test datasets with 0.67 and 0.33 respectively as size.

1. Decision Tree
2. Random Forest Classifier
3. Multilayer Perceptron's (MLPs)
4. XGBoost Classifier
5. Support Vector Machines

### Model-1: Decision Tree Classifier and Results

Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation.

DecisionTreeClassifier is a class capable of performing multi-class classification on a dataset. As with other classifiers, DecisionTreeClassifier takes as input two arrays: an array X, sparse or dense, of shape (n_samples, n_features) holding the training samples, and an array Y of integer values, shape (n_samples,), holding the class labels for the training samples.

After being fitted, the model can then be used to predict the class of samples. In case that there are multiple classes with the same and highest probability, the classifier will predict the class with the lowest index amongst those classes. As an alternative to outputting a specific class, the probability of each class can be predicted, which is the fraction of training samples of the class in a leaf. DecisionTreeClassifier is capable of both binary (where the labels are [-1, 1]) classification and multiclass (where the labels are [0, …, K-1]) classification.

We have used the Decision Tree Classification model from the sklearn library and trained the model with a max depth of 5. We used the model on the test data and predicted the result. The decision tree has an accuracy of almost 80.2 % on predicting the result based on the test data, when compared to the actual data. We Observed that the URL Length was the most important feature for classification.

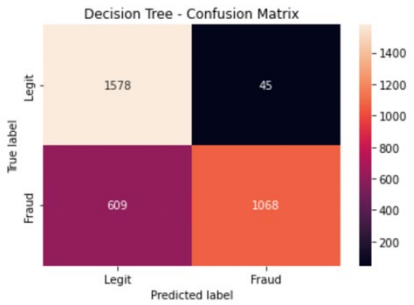



*Figure: Decision Tree Confusion Matrix*

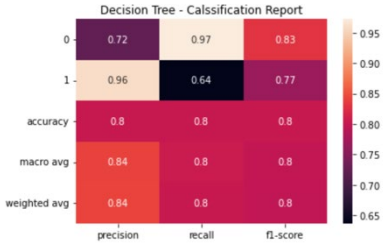



*Figure: Decision Tree Classification Report*

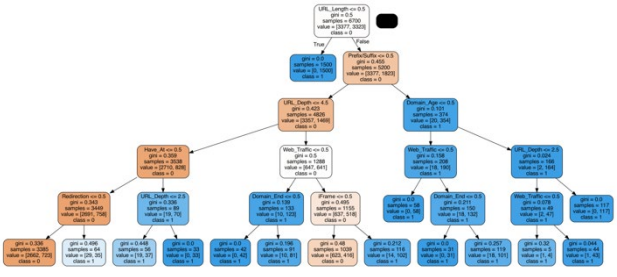


*Figure: Decision Tree Structure*

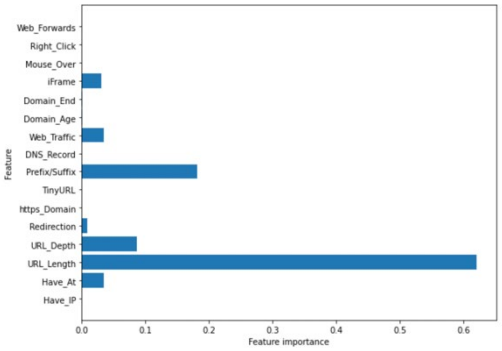



*Figure: Decision Tree – Feature Importance*

**Model-2: Random Forest Classifier and Results**

The Random Forest is an ensemble method, which uses a collection of a large number of decision trees to make predictions instead of individual models. The individual models must make predictions which are independent of each other, and each individual model should be better than a random Classifier.

The default values for the parameters controlling the size of the trees (e.g. max_depth, min_samples_leaf, etc.) lead to fully grown and unpruned trees which can potentially be very large on some data sets. To reduce memory consumption, the complexity and size of the trees should be controlled by setting those parameter values.

The features are always randomly permuted at each split. Therefore, the best found split may vary, even with the same training data, max_features=n_features and bootstrap=False, if the improvement of the criterion is identical for several splits enumerated during the search of the best split. To obtain a deterministic behaviour during fitting, random_state has to be fixed.

We have used the Random Forest Classification model from the sklearn library and trained the model with a max depth of 5. We used the model on the test data and predicted the result. The Random Forest Classification has an accuracy of almost 81.2 % on predicting the result based on the test data, when compared to the actual data. We can also observe that the URL Length was the most important feature for classification.

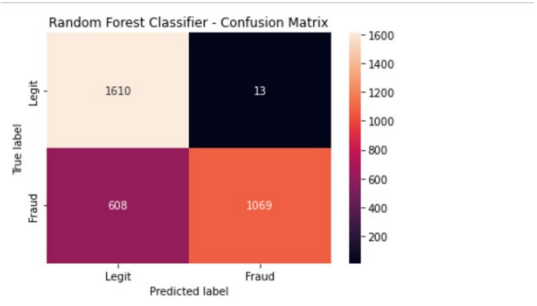



*Figure: Random Forest Classifier Confusion Matrix*

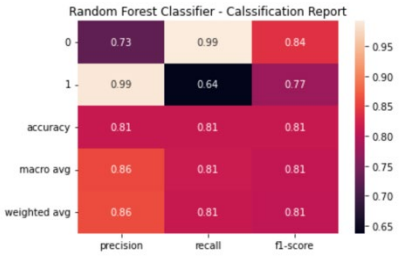



*Figure: Random Forest Classifier Classification Report*

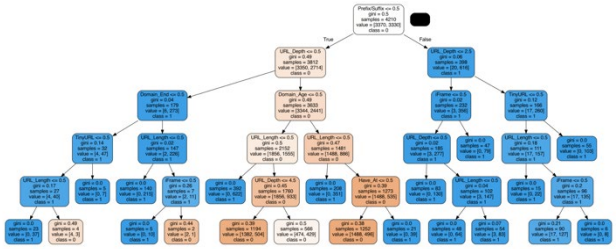


*Figure: Random Forest Classifier Tree Structure*

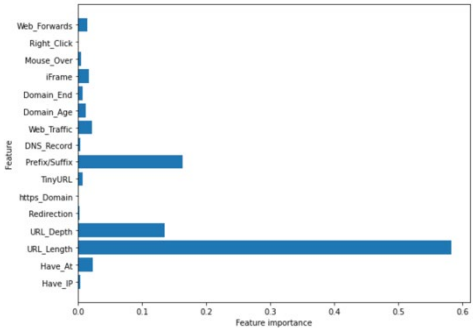



*Figure: Random Forest Classifier Feature Importance*

### Model-3: Multi-Layer Perceptron's and Results

Multi-layer Perceptron (MLP) is a supervised learning algorithm that learns a function f(·):R^m→R^o by training on a dataset, where m is the number of dimensions for input and o is the number of dimensions for output. Given a set of features X=x1,x2,...,xm and a target y, it can learn a non-linear function approximator for either classification or regression. It is different from logistic regression, in that between the input and the output layer, there can be one or more non-linear layers, called hidden layers. Figure 1 shows a one hidden layer MLP with scalar output.

The leftmost layer, known as the input layer, consists of a set of neurons {xi|x1,x2,...,xm} representing the input features.

Each neuron in the hidden layer transforms the values from the previous layer with a weighted linear summation w1x1+w2x2+...+wmxm, followed by a non-linear activation function g(·):R→R - like the hyperbolic tan function. The output layer receives the values from the last hidden layer and transforms them into output values.

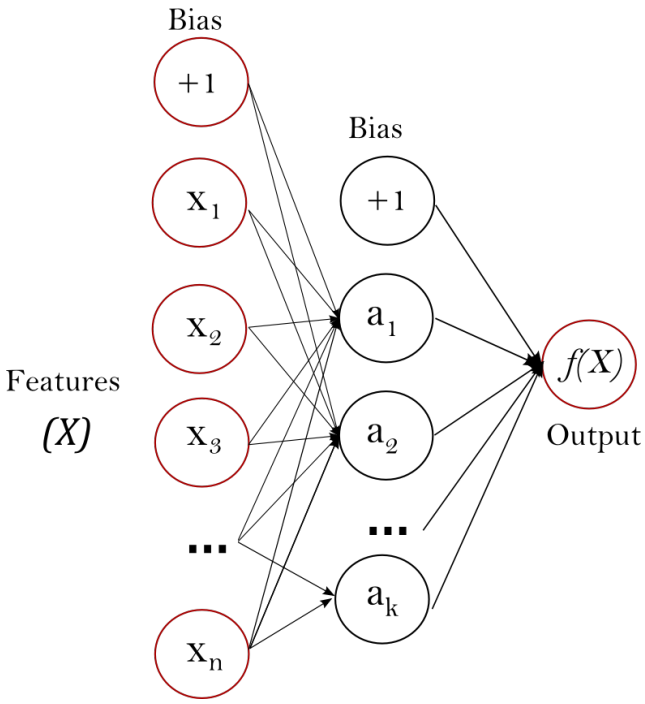



*Figure: One Hidden Layer Multi-Layer Perceptron.*

We have used the multilayer perceptron artificial neural network from the sklearn.neural_network and we have used the MLP Classifier to classify the urls into legitimate and phishing by training the model and then predicting the result of the test data. The accuracy of the model was found to be 84.6% better than both the decision tree and the random forest classification models.

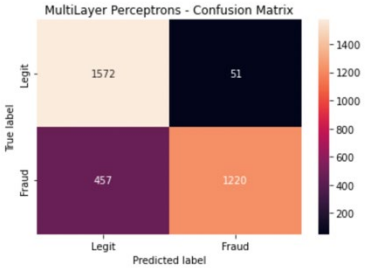



*Figure: MLP Classifier Confusion Matrix*

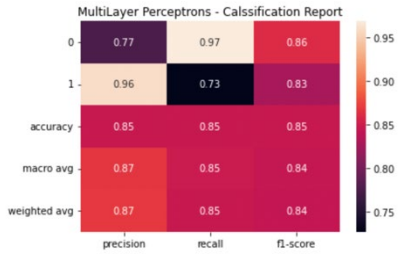



*Figure: MLP Classifier Classification Report*

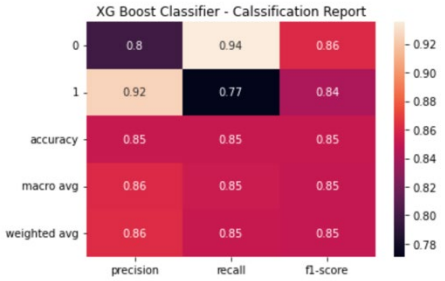



*Figure: XGBoost Classifier Classification Report*

**Model-4: XGBoost Classifier and Results**

XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible, and portable. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost provides a parallel tree boosting (also known as GBDT, GBM) that solve many data science problems in a fast and accurate way. The same code runs on major distributed environment (Hadoop, SGE, MPI) and can solve problems beyond billions of examples.

XGBoost 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.

We have used the XGBoost Classifier from the xgboost library to classify the data. We have set the max depth as 7 and the learning rate as 0.4.

The model was trained using the test data and the result predicted for the test data had an accuracy of 85.1 % which was better than that of the above-mentioned classifiers.

**Model-5: Support Vector Machines and Results**

Support Vector Machines, or SVM, are machine learning models capable of performing linear or nonlinear classification, regression, and outlier detection. For linear classifica- tion, SVM splits data by maximizing the margin between classes. This results in outliers not drawing the decision boundary away, so it is fully determined (or" supported") by the instances located closest to the boundary. These instances are called the support vectors.

For classification via the standard SVM methodology, the penalty weight assigned to misclassifications are the same for every datapoint. Introducing variable weights into the loss function can help to pull the decision boundary away from the fraudulent cases at the expense of a few non-fraudulent misclassifications.

SVM is a supervised machine learning algorithm which uses a technique called the kernel trick to transform your data and then based on these transformations it finds an optimal boundary between the possible outputs.

We have used the support vector machine model from the sklearn library to classify the data into phishing or legitimate. We tested the support vector classifier with a linear kernel and had an accuracy of 79.96% which is lesser than all the remaining classifiers.

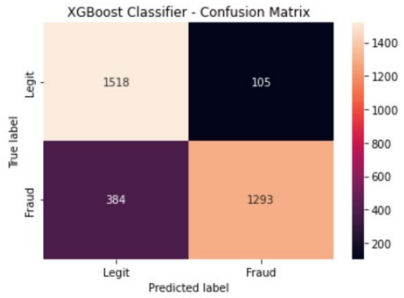



*Figure: XGBoost Classifier Confusion Matrix*

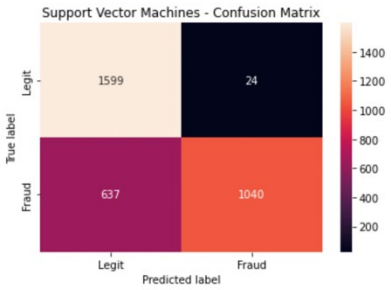



*Figure: SVM Confusion Matrix*

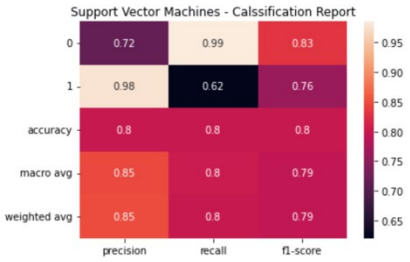



*Figure: SVM Classification Report*

## 8. Comparing the Model Results

We worked on 5 models and corresponding accuracies were displayed in the below figures including an accuracy plot. XGBoost outperformed compared to all the other models followed by MLP with a very slight difference and we observed that SVM is least performed model for this dataset. All the confusion matrix for each model and the classification report including precision, recall and f1score is displayed along with the models.

Decision Tree Train and Test Accuracies:

```
Decision Tree:
Accuracy on training Data: 0.8162686567164179
Accuracy on test Data: 0.8018181818181818
```

Random Forest Classifier Train and Test Accuracies:

```
Random forest:
Accuracy on training Data: 0.8240298507462687
Accuracy on test Data: 0.8118181818181818
```

Multilayer Perceptron's Train and Test Accuracies:

```
Multilayer Perceptrons:
Accuracy on training Data: 0.8655223880597015
Accuracy on test Data: 0.8460606060606061
```

XGBoost Classifier Train and Test Accuracies:

```
XGBoost:
Accuracy on training Data: 0.8714925373134328
Accuracy on test Data: 0.8518181818181818
```

SVM Train and Test Accuracies:

```
SVM:
Accuracy on training Data: 0.8029850746268656
Accuracy on test Data: 0.7996969696969697
```

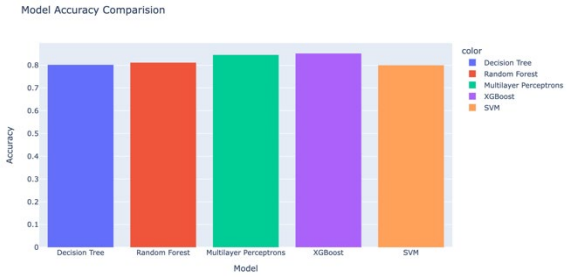



*Figure: Model Comparison Accuracy Plot*

## 9. Hyper Parameter Tuning (Decision Tree):

XGBoost is the best classifier model with highest accuracy when compared to other models and MLP with very slight difference almost equals XGBoost. Now we decided to perform hyperparameter tuning for Decision tree and will compare the results with other models.

Initial Decision tree is performed with default parameters with max_depth as 5 , criterion (gini) and without specifying the min samples leaf. Now we are performing hyperparameter tuning by checking the best decision tree model with criterion as gini and entropy, max_depth as [2,3,5,10,20] and min_samples_leaf as [5, 10,20,50,100]. We used GridSearchCV from sklearn model selection and achieved the best decision tree model with parameters as criterion = entropy, max_depth = 20 and min_samples_leaf = 5. Using this model we derived the confusion matrix, classification report and reported the accuracy.

Accuracy we got after performing hyper parameter tuning for decision tree is 84.9% which is almost nearby to MLP and XGBoost.

```
DecisionTreeClassifier
DecisionTreeClassifier(criterion='entropy', max_depth=20, min_samples_leaf=5,
                       random_state=29)
```

*Figure: Best Decision Tree Model Parameters*

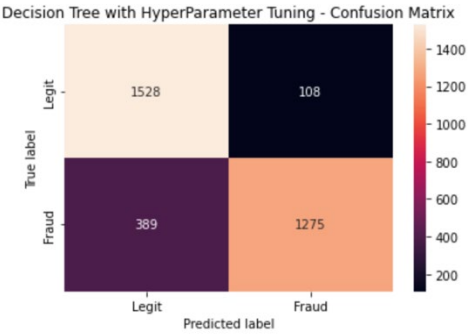



*Figure: Decision Tree with Hyper Parameter Tuning Confusion Matrix*

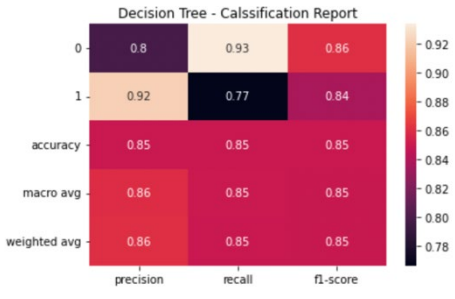



*Figure: Decision Tree with Hyper Parameter Tuning Confusion Matrix*

Decision Tree with hyperparameter tuning Train and Test Accuracies:

```
Decision Tree with HyperParameter Tuning:
Accuracy on training Data: 0.861044776119403
Accuracy on test Data: 0.8493939393939394
```

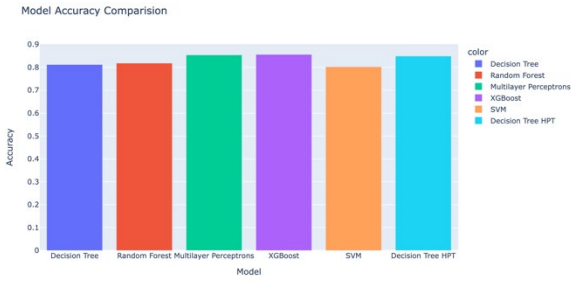



*Figure: Model Comparison Accuracy Plot including Decision Tree with Hyperparameter Tuning*

## 10. Conclusions

In this paper, we present Decision Tree, Random Forest, Multilayer Perceptron, XGBoost and SVM classifier models to detect fraudulent website on a dataset of legit and fraudulent websites. XGBoost outperformed with highest accuracy followed by Multilayer Perceptron's with a very slight difference and SVM performed least with less accuracy compared with other models. After performing Hyperparameter tuning for Decision tree model the accuracy achieved is almost nearby to XGBoost and MLP with very slight difference. However, on a different dataset, potentially the MLP may be better suited and perform better.

## References

https://en.wikipedia.org/wiki/Phishing
https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0258361
https://www.phishtank.com/developer_info.php
https://www.unb.ca/cic/datasets/url-2016.html
https://scikit-learn.org/stable/modules/tree.html
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html
https://xgboost.readthedocs.io/en/stable/
https://scikit-learn.org/stable/modules/svm.html
https://towardsdatascience.com/how-to-tune-a-decision-tree-f03721801680

## Annex:

https://github.com/rajeshchinnaga/MachineLearning/blob/main/IML-Project/IML-Project-TrainingModels.ipynb

IML Project Folder has all the code including datasets.