International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 10 Issue: 07 | July 2023 www.irjet.net p-ISSN: 2395-0072

Machine Learning-Based Phishing Detection

Mohammed Naif[1], Allen Jeriel K[2], Sneh Patil[3], Ananya Rangaraju[4], Peraka Divyanjali[5], Kandukuri Pavan Sai Praveen[6]

Abstract - *Millions of users have been successfully connected globally by the internet today, and as a result, users' reliance on this platform for data browsing, online transactions, and information downloads has grown. Cybersecurity is a term for a collection of technologies and procedures used to safeguard software and hardware against intrusion, harm, and attacks. DoS attacks, Man-in-the-Middle attacks, Phishing attacks, SQL Injection attacks, etc. are some of the most often seen cybersecurity threats. There has been an uptick in consumers losing access to their very sensitive and private information over the past few years. These days, fraudsters utilise such methods to trick their victims in an effort to steal personal information including their username, password, bank account information, and credit card information. Attacks against users are frequently delivered via spoofing emails, illegal websites, malware, etc. To handle complicated and massive amounts of data, a structured automated technique is necessary. The most common and effective approach that can be used to address this issue is machine learning, according to research. The most widely used machine learning methods include neural networks, decision trees, logistic regression, and support vector machines (SVM). A group of deep learning and machine learning models will be trained in this study to identify phishing websites.*

Key Words: **Machine Learning, Cyber Security, Phishing, Neural Network, Website Security**

1.INTRODUCTION

The majority of our daily activities, including shopping and banking, have been moved online thanks to the network. A network that is unregulated or unprotected serves as a launchpad for a variety of cyberattacks, creating major security risks not only for networks but also for regular computer users, even seasoned ones. The users must be protected from these intrusions, which is crucial. Phishing website attacks are among the most frequent types of cyberattacks. A phishing website is a popular social engineering technique that imitates reliable URLs and web sites. The phisher targets unsuspecting web users through such attacks in an effort to deceive them into disclosing private information in order to use it fraudulently. Due to the end user's weakness, an attacker can even utilise new approaches to target those seasoned users who have already provided personal information while believing that the page is real. Software-based phishing detection solutions are therefore recommended as user decision support tools. Machine learning algorithms and techniques are used in this project. The goal of this research is to use the dataset produced to predict phishing websites to train machine learning models and deep neural networks. In order to create a dataset from which the necessary URL- and website content-based attributes may be extracted, both phishing and benign URLs of websites are collected. Each model's performance level is assessed and contrasted. Before feeding into the algorithms, the feature extraction will be carried out based on the address bar, domain-based, HTML, and Javascript extraction. In this project, machine learning methods like Decision Trees, Random Forests, Multilayer Perceptrons, XGBoost, and Support Vector Machines will be used. The models will be assessed, and accuracy will be a factor in the evaluation. Since XGBoost features built-in L1 (Lasso Regression) and L2 (Ridge Regression) regularisation, which prevents the model from overfitting, the algorithm should perform better in this situation. Additionally, XGBoost has the capacity to handle missing numbers right out of the box. When a node has a missing value, XGBoost attempts both the left and right hand splits and learns which one results in the highest loss for each node. The website's security is a vulnerability in this project because there isn't enough security on the site, which leaves it open to cyberattacks.

1.1 Novelty

Our suggested methodology, which considers not only the URL-based features of phishing websites but also their Domain-based features, as well as the HTML and Java-script based features during feature extraction, aims to reduce the False Positive Rate as well as the False Negative Rate and improve overall accuracy. The model may be trained to recognise phishing sites that substitute textual content with embedded objects like flash, java scripts, and HTML files by applying this additional set of features.

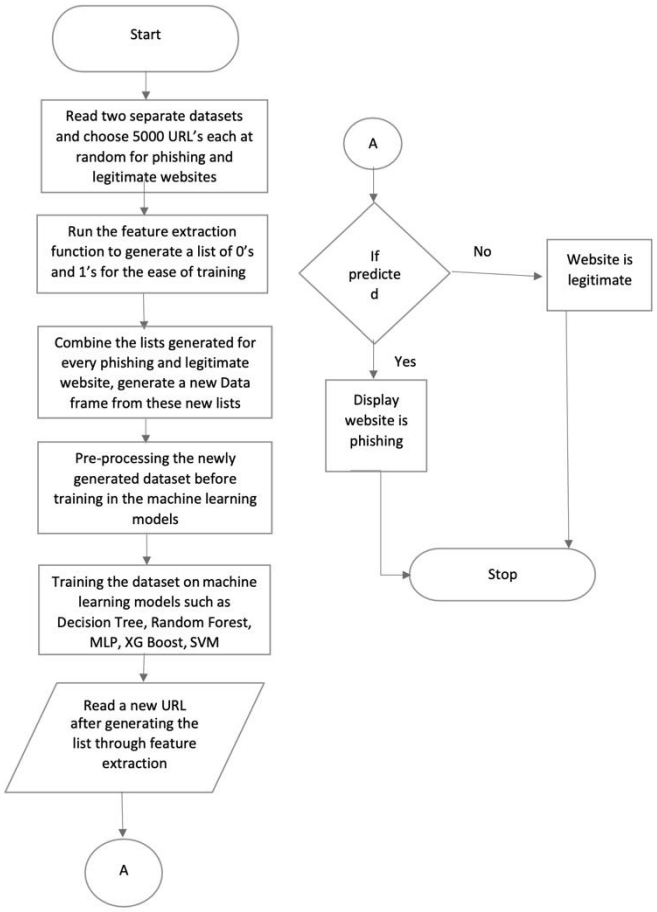
2.ALGORITHM

1. Compile a dataset from open source platforms that includes both phishing and trustworthy websites.
2. Take the necessary information out of the URL database.
3. Utilise EDA techniques to analyse and pre-process the dataset.
4. Create training and testing sets from the dataset.
5. Run a few deep learning and machine learning algorithms, such as SVM, Random Forest, and Autoencoder.

6. Create a programme that displays the evaluation outcome while taking accuracy metrics into account.

7. Compare the outcomes obtained using the trained models, and state which is superior.

3.FLOW CHART

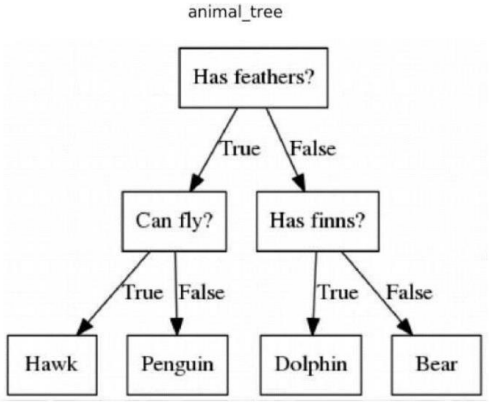


4.ML MODELS USED

This data collection has a categorization issue because the input URL can either be considered legal (0) or phishing (1). The following machine learning models (classification) were taken into account during training the dataset:

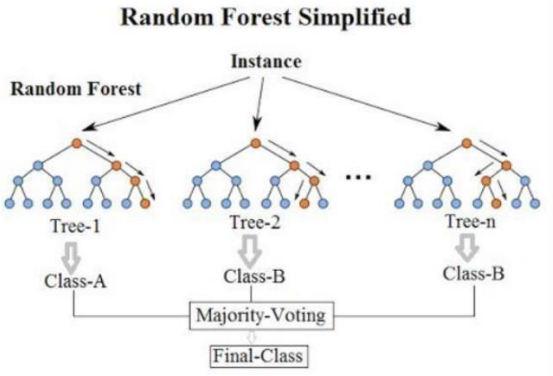
4.1 Decision Tree

Decision trees are a type of predictive modelling that can be used to map several options or solutions to a certain result. Different nodes make up decision trees. The decision tree's root node, which in machine learning typically represents the entire dataset, is where it all begins. The leaf node is the branch's termination point or the result of all previous decisions. From a leaf node, the decision tree won't branch out any further. In machine learning decision trees, the internal nodes represent the data's features, while the leaf node represents the result.



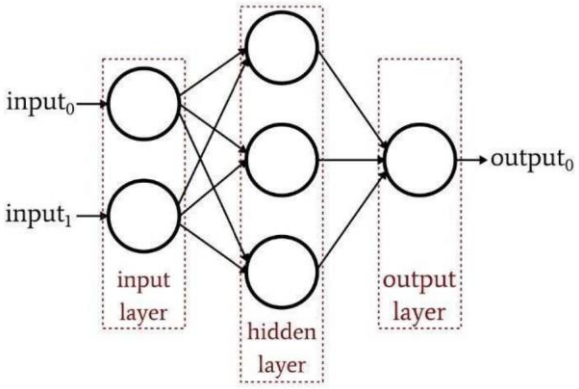
4.2 Random Forest

The most well-known and effective supervised machine learning approach is called the Random Forest approach. The Random Forest Algorithm can handle both classification and regression tasks. This approach builds a Forest with a number of Decision Trees, as the term "Random Forest" suggests. In general, a prediction is more accurate and strong the more trees there are in the forest. The Random Forest Algorithm creates a forest out of several decision trees. The process used to build a Single Decision Tree is also utilised to build a Decision Tree. To create a decision tree, the information gain and entropy are calculated.



4.3 Multilayer Perceptron

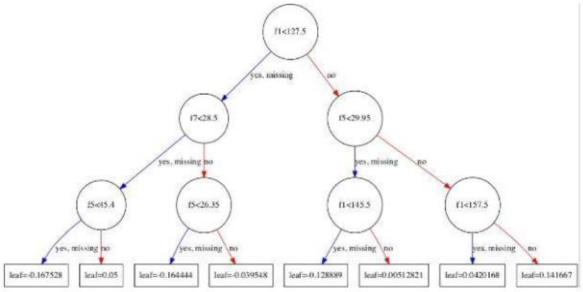
A type of feedforward artificial neural network (ANN) is called a multilayer perceptron (MLP). The name "MLP" is unclear; it can refer to any feedforward ANN in some contexts, or it can specifically refer to networks made up of multiple layers of perceptrons (with threshold activation) in others. When multilayer perceptrons have just one hidden layer, they are sometimes referred to as "vanilla" neural networks.



4.4 XG Boost

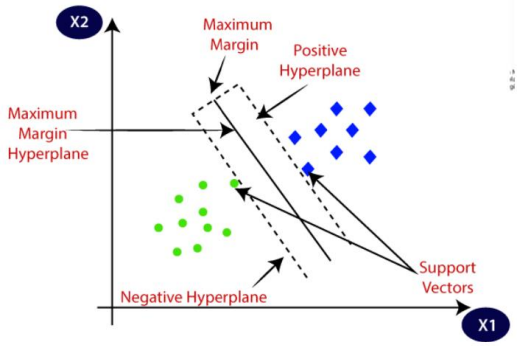
Extreme Gradient Boosting is a machine learning approach that depends on a decision tree and is used to solve regression and classification problems. In order to reduce the errors of the preceding tree, this algorithm generates decision trees one at a time. Every tree learns from its predecessor and updates the leftover mistakes. As a result, each subsequent tree in the series continues to learn from the mistakes made by the prior branch.

The 'weak learners' in XGBoost are the base learners, where the bias is substantial and the predictive capacity is just slightly higher than random guessing. Because each of the weak learners provides essential data for estimation, the boosting strategy can combine the weak learners to produce a strong learner. The final effective learner eliminates bias and variation.



4.5 Support Vector Machines

In a high- or infinite-dimensional space, a support-vector machine creates a hyperplane or set of hyperplanes that can be used for classification, regression, or other tasks like outliers detection. Generally speaking, the higher the margin, the smaller the classifier's generalisation error, so a decent separation is obtained by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin).



5. MODEL EVALUATION

When evaluating the models, accuracy is taken into account. The accuracy of each of the ML algorithms mentioned above on the given data set will be calculated, and the one with the best accuracy will be saved and used for the model's further deployment.

The percentage of correctly identified cases for (TP + TN)/(TP + TN + FP + FN) can be used to measure accuracy. where the letters TP, FN, FP, and TN stand for the corresponding numbers of true positives, false negatives, FPs, and TNs.

The correct classifications are true positives (TP) and true negatives (TN). When an algorithm's result is mistakenly anticipated even though it is actually present in the image, it is referred to as a false positive (FP).

- 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

- The dataset of phishing websites has 5000 URLs that were randomly selected.

```
#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...

5. Machine Learning Models & Training

5.1 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)
```

4.2 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)
```

5.3 Multilayer Perceptron

```
# 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)
```

5.4 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)
```

5.5 Support Vector Machines

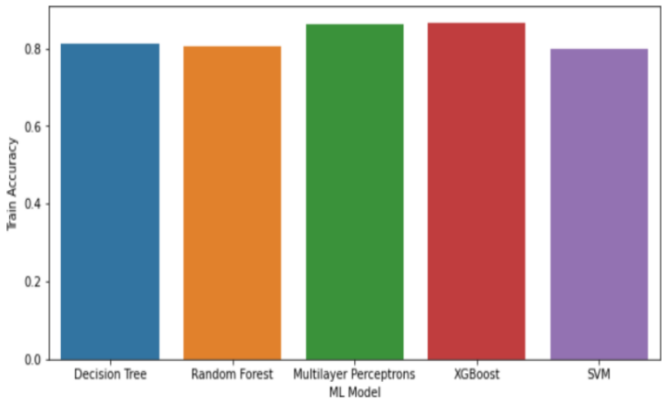
```
#Support vector machine model
from sklearn.svm import SVC

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

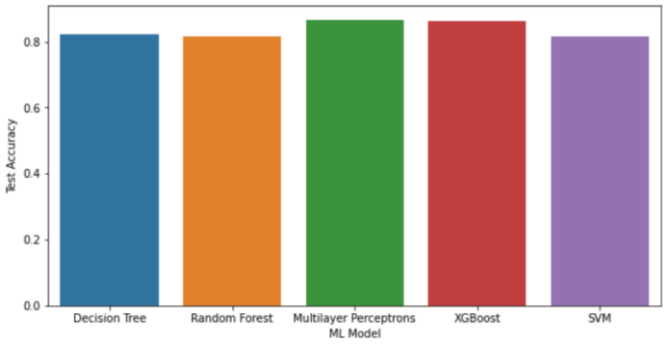
6. Comparison of Models

We compared the models. It is evident from the comparison that the XGBoost Classifier performs well with this dataset.

Test Accuracy:



Train Accuracy:



7. RESULTS OBTAINED

To see if the model is effective, we test it using a well-known phishing website.

```
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 1 ):
  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.

7. CONCLUSION

One of the best and simplest ways for hackers to defraud us of our money and obtain our financial and personal information is still phishing. Machine learning algorithms can learn to differentiate between trustworthy and fraudulent websites by being trained on a sizable dataset of both. This may result in the creation of efficient phishing detection tools that can instantly recognise and alert users to potentially hazardous websites. We have determined that XGBoost Classifier is the best method for this particular application after researching several ML algorithms. Even though employing machine learning for phishing detection has numerous advantages, there are some restrictions and difficulties that need to be resolved. Making sure the algorithms can recognise new and evolving phishing attack types is one of the major problems. The training data and algorithmic feature sets must be continuously updated for this. Additionally, adversarial assaults, in which attackers alter the characteristics of phishing websites to avoid detection, might expose machine learning algorithms to risk. Using machine learning models that are secure and resistant to these assaults is crucial to addressing this.

REFERENCES

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

[2] Thomas Rincy N 1 and Roopam Gupta, Design and Development of an Efficient Network Intrusion Detection System Using Machine Learning Techniques, Hindawi Wireless Communications and Mobile Computing Volume 2021, Article ID 9974270, 35

pages https://doi.org/10.1155/2021/9974270

[3] Gadge, Jayant; Patil, Anish Anand (2008). [IEEE 2008 16th IEEE International Conference on Networks - New Delhi, India (2008.12.12-2008.12.14)] 2008 16th IEEE International Conference on Networks - Port scan detection.

[4] Vasilomanolakis, Emmanouil; Sharief, Noorulla; Muhlhauser, Max (2017). [IEEE 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM) - Lisbon, Portugal (2017.5.8-2017.5.12)] 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM) - Defending against Probe-Response Attacks. , (), 1046–1051. doi:10.23919/inm.2017.7987436

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

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