International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 12 | December - 2022 Impact Factor: 7.185 ISSN: 2582-3930

DETECTION OF PHISHING WEBSITES USING ML

T. ABHIRAM, K. CHAITHANYA
D.SREENIVAS RAO (Guide)
Sreenidhi Institute of Science and Technology
Hyderabad

ABSTRACT:

Phishing is one of the familiar attacks that trick users to access malicious content and gain their information. In terms of website interface and uniform resource locator (URL), most phishing webpages look identical to the actual webpages. Various strategies for detecting phishing websites, such as blacklist, heuristic, Etc., have been suggested. However, due to inefficient security technologies, there is an exponential increase in the number of victims. The anonymous and uncontrollable framework of the Internet is more vulnerable to phishing attacks. Existing research works show that the performance of the phishing detection system is limited. There is a demand for an intelligent technique to protect users from the cyber-attacks. In this study, we proposed a URL detection technique based on machine learning approaches. A recurrent neural network method is employed to detect phishing URL.

INTRODUCTION

The Internet has become an important part of our lives for gathering and disseminating information, particularly through social media. According to Pamela (2021), the Internet is a network of computers containing valuable data, so there are many security mechanisms in place to protect that data, but there is a weak link: the human. When a user freely gives away their data or access to their computer, security mechanisms have a much more difficult time protecting their data and devices. Therefore, Imperva (2021) defines social engineering (a type of attack used to steal user data, including login credentials and

credit card numbers) as a type of attack that is one of the most common social engineering attacks. The attack happens when an attacker fools a victim into opening an email, instant message, or text message as if it were from a trusted source. Upon clicking the link, the recipient is fooled into believing that they've received a gift and unsuspectingly clicks a malicious link, resulting in the installation of malware, the freezing of the system as part of a ransomware attack, or the disclosure of sensitive information. The strategies employed by cybercriminals are becoming more complex as technology advances. Other than phishing, there are a variety of methods for obtaining personal information from users. KnowBe4 (2021) stated the following techniques: a) Vishing (Voice Phishing): This kind of phishing includes the phisher calling the victim to get personal information about the bank account. The most common method of phone phishing is to use a phony caller ID. b) Smishing (SMS Phishing): Phishing via Short Message Service (SMS) is known as Smishing. It is a method of luring a target through the SMS text message service by sending a link to a phishing website. c) Ransomware: A ransomware attack is a type of attack that prevents users from accessing a device or data unless they pay up. d) Malvertising: Malvertising is malicious advertising that uses active scripts to download malware or push undesirable information onto your computer. The most prevalent techniques used in malvertisements are exploits in Adobe PDF and Flash. Hence, this is a rapidly evolving threat to individuals as well as big and small corporations. Criminals now have access to industrial-strength services on the dark web, resulting in an increase in the amount of these phishing links and emails, and, more frighteningly, they are increasing in ‘quality,' making them tougher to detect.

Fig 1: Protection against Cyber Attacks

WHAT IS 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.

Objective:

Phishing website is one of the internet security problems that target the human vulnerabilities rather than software vulnerabilities. It can be described as the process of attracting online users to obtain their sensitive information such as usernames and passwords.The objective of this project is to train machine learning models and deep neural network on the dataset created to predict phishing websites. Both phishing and legitimate 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.

Phishing Detection Model:

The based methodology stated that the proposed system utilizes machine learning models and deep neural networks. These models consist of Decision Tree, Support Vector Machine, XGBooster, Multilayer Perceptions, Auto Encoder Neural Network, and Random Forest. The models determine whether a website URL is phishing or legitimate. The models help give a 2-class prediction (legitimate (0) and phishing (1)). In the model development process, over six (6) machine learning models and deep neural network algorithms all together were used to detect phishing URLs using Jupyter notebook IDE with packages such as pandas, Beautiful Soup, who-is, urllib, etc. Here are the models, their accuracy was tested using sklearn matrices with an accuracy score and their matrices are shown in figure 4.13. The XGBooster model had the highest performance score of 86.6%, the Multilayer Perceptions model had an accuracy of 86.5%, the Decision Tree model had an accuracy of 81.4%, the Random Forest model had an accuracy of 81.8%, the Support Vector Machine model had an accuracy of 80.4%, and the Auto Encoder Neural Network model had an accuracy of 16.1%.

	ML Model	Train Accuracy	Test Accuracy
3	XGBoost	0.866	0.864
2	Multilayer Perceptrons	0.865	0.864
0	Decision Tree	0.814	0.812
1	Random Forest	0.818	0.811
5	SVM	0.804	0.794
4	AutoEncoder	0.161	0.177

Model Development:

The model for detecting phishing URL websites was built using a python programming language with over six (6)

machine learning models and deep neural network algorithms altogether and the most accurate test score on the tested 5,000 datasets were used.

Data Collection:

The dataset used for the classification was sourced from was gotten from multiple sources listed in the earlier stated methodology. The dataset used for classifying the dataset into phishing and legitimate URLs was sourced from open source websites.

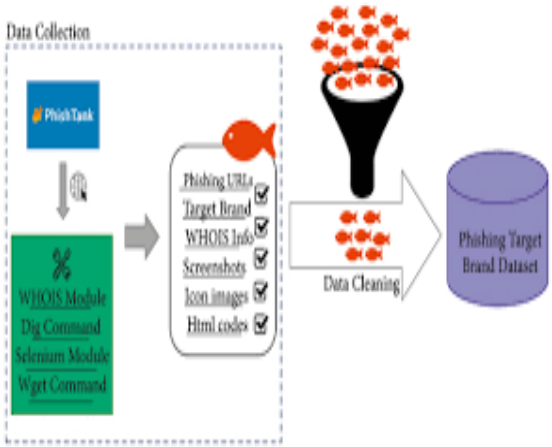


Fig2: Data Collection

The set of phishing URLs are collected from opensource service called "Phish Tank". This service provide a set of phishing URLs in multiple formats like csv, json etc. that gets updated hourly. To download the data:

https://www.phishtank.com/developer_info.php. From this dataset, 5000 random phishing URLs are collected to train the ML models.

****legitimate URL Dataset****

The legitimate URLs are obatined from the open datasets of the University of New Brunswick, https://www.unb.ca/cic/datasets/url-2016.html. This dataset has a collection of benign, spam, phishing, malware & defacement URLs. Out of all these types, the benign url dataset is considered for this project. From this dataset, 5000 random legitimate URLs are collected to train the ML models.

Models & Training

Before stating the ML model training, the data is split into 80-20 i.e., 8000 training samples & 2000 testing samples. From the dataset, it is clear that this is a supervised machine learning task. There are two major types of supervised machine learning problems, called classification and regression.

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 project are:

* Decision Tree

* Random Forest

* Multilayer Perceptrons

* XGBoost

* Autoencoder Neural Network

* Support Vector Machines

All these models are trained on the dataset and evaluation of the model is done with the test dataset.

The model XGBoost has the highest accuracy and also The listed (feature extraction) to validate URLs were integrated to a webapplication using django framework to effeciently detect phishing URL links.

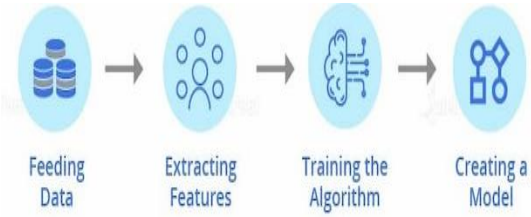


Fig 3: Training the Model

Data Analysis & Visualization

The image as shows the distribution plot of how legitimate and phishing datasets are distributed base on the features selected and how they are related to each other. The figure shows the plot of a correlation heat-map of the dataset. The plot shows correlation between different variables in the dataset. In figure 4.9 and figure 4.10, it shows the feature importance in the model for Decision tree classifier and Random forest classifier respectively.

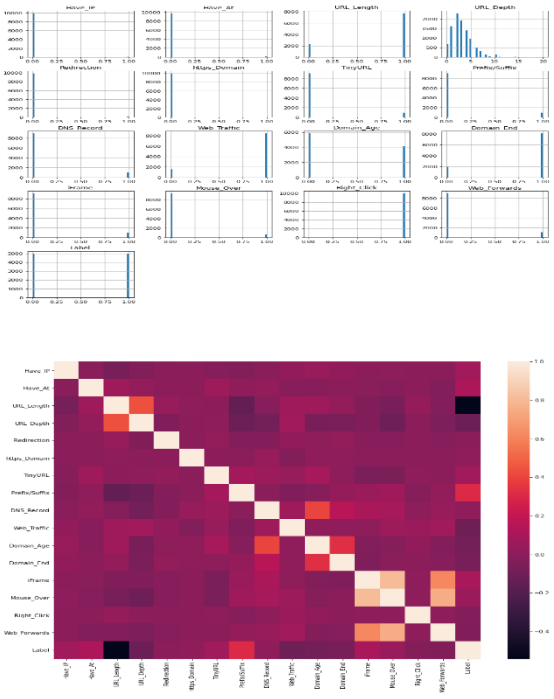


Fig4 : Visualization

STEPS FOR IMPLEMENTATION

- Data Preprocessing
- Importing Libraries
- Feature Extraction
- Training Machine Learning Models
- Results

1.Data Preprocessing:

The datasets were first cleaned to remove empty entries and fill some entries by applying data pre-processing techniques and transform the data to use in the models. Figure 4.11 shows the summary of the dataset while figure 4.12 shows the number of missing values in the dataset which all appear to be zero.

Fig5: DataSet of this Project

Source: The Dataset is collected from an open-source service called Phish-Tank. This dataset consists of 5,000 random phishing URLs which are collected to train the ML models.

2.IMPORT LIBRARIES

In this project, we're utilizing a variety of libraries Pandas,Tensorflow, Matplotlib, NumPy,seaborn, Scikit-learn, and Keras.

Pandas: It is an open source Python package that is most widely used for data science/data analysis and machine learning tasks. It is built on top of another package named Numpy, which provides support for multi-dimensional arrays

Tensorflow: It is a Machine Learning and AI library with a wide range of algorithms. The main focus of this library is to train models and interface with deep neural networks.

Matplotlib: It is a visualization library that is used to present the results in more understandable ways i.e., in graphs and plots.

Seaborn:

It is a library that uses Matplotlib underneath to plot graphs. It will be used to visualize random distributions..

3.Feature Extraction

The below mentioned category of features are extracted from the URL data:

1. Address Bar based Features

- In this category 9 features are extracted.

2. Domain based Features

- In this category 4 features are extracted.

3. HTML & Javascript based Features

- In this category 4 features are extracted.

So, all together 17 features are extracted from the 10,000 URL dataset and are stored in file in the DataFiles folder.

4.Training Machine Learning Models :

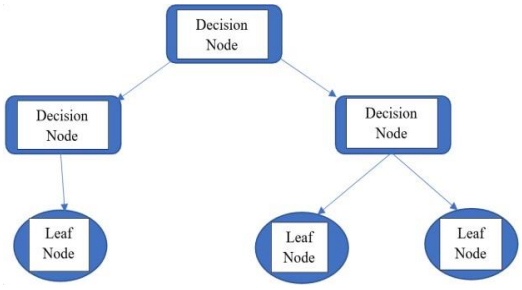
From the dataset above, it is clear that this is a supervised machine learning task. There are two major types of supervised machine learning problems, called classification and regression.

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:

- Decision Tree
- Random Forest
- Multilayer Perceptrons
- XGBoost
- Autoencoder Neural Network
- Support Vector Machines
-

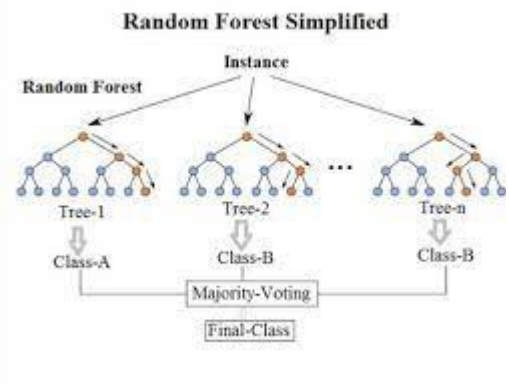
- **Decision Tree Classifier:**

- Decision Tree Classifiers are widely used in classification and regression tasks which involvea decision task such as if/else question. This is an optimal decision maker that couldgive us the best decision much quicker.
-
- In ML models, these decisions are named as tests. Tests in the sense, to find out whether our model is generalizable or not. This algorithm can be used to sequenciate all the tests fromthe dataset to find the information about the target variable.



- Fig5:Decision Tree

- **Random Forest Classifiers:**
- It can be defined as a collection of decision trees. Here, multiple number of decision trees are collected together and worked simultaneously to get the best of the average of the result. This will be very robust and all you need to know no of trees required before building a random forest. This does not any parameters or scaling of data, all you need is n_estimators.



- **Fig4**: Random Forest

Multilayer Preceptrons:

Multilayer Perceptrons are known as feed forward neural networks. They are used to process multiple stages simultaneously and result in an optiomal decision for the processed stage. A multilayer perceptron is a fully connected class of feedforward artificial neural network. The term MLP is used ambiguously, sometimes loosely to mean any feedforward ANN, sometimes strictly to refer to networks composed of multiple layers of perceptrons.

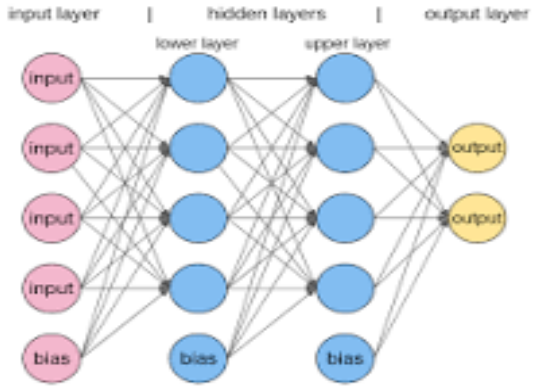


Fig7: MultiLayer Perceptron

XGBoost :

XGBoost is an open-source Python library that provides a gradient boosting framework. It helps in producing a highly efficient, flexible, and portable model. When it comes to predictions, XGBoost outperforms the other algorithms or machine learning frameworks. This is due to its accuracy and enhanced performance.

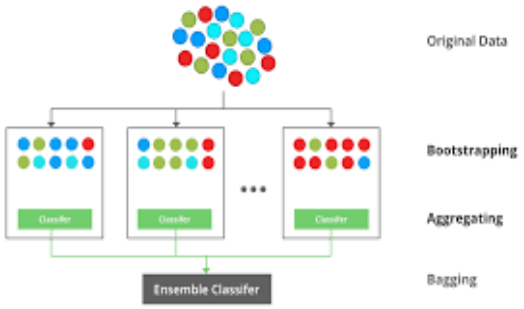


Fig8 :XGBoost

SVM:

SVMs are used in applications like handwriting recognition, intrusion detection, face detection, email classification, gene classification, and in web pages. This is one of the reasons we use SVMs in machine learning. It can handle both classification and regression on linear and non-linear data.

RESULTS

There are primarily three modes of phishing detection²: Content-Based Approach: Analyses text-based content of a page using copyright, null footer links, zero links of the body HTML, links with maximum frequency domains. Using only pure TF-IDF algorithm, 97% of phishing websites can be detected with 6% false positives.

Fig: Results showing Accuracy

```
#creating dataframe
results = pd.DataFrame({ 'ML Model': ML_Model,
                         'Train Accuracy': acc_train,
                        'Test Accuracy': acc_test})
results
```

	ML Model	Train Accuracy	Test Accuracy
0	Decision Tree	0.814	0.812
1	Random Forest	0.818	0.811
2	Multilayer Perceptrons	0.865	0.864
3	XGBoost	0.866	0.864
4	AutoEncoder	0.161	0.177
5	SVM	0.804	0.794

Fig: Results

MERITS

When it comes to merits, the higher accuracy of our project makes it more reliable for decision-making. It helps to detect the phishing website and prevents lose of data.

LIMITATIONS

The process of this project is highly reliable but it requires high-processing computers. Normal computers cannot handle the processing rate of this code. It requires a

high graphic processing rate to get a higher accuracy of the data.

SOFTWARE REQUIREMENTS:

The software requirements for the development of this system include: i. Windows Operating System (8/10) ii. Anaconda Navigator (Jupyter Notebook) iii. PyCharm Community edition iv. Web browser (Preferably Chrome) v. Visual Studio Code

HARDWARE REQUIREMENTS:

The hardware requirement includes: i. A laptop or desktop computer (Preferably 64bit) ii. Random Access Memory (RAM): 8 Gigabytes Minimum iii. Processor: Intel Core i5, 2.4 GHz Minimum

CONCLUSION

Phishing campaigns can be difficult spot. Cyber criminals have become experts at using sophisticated techniques to trick victims into sharing personal or financial information.

But the best way to protect yourself is to learn how to spot a phishing scam before you take the bait.

The system developed detects if a URL link is phishing or legitimate by using machine learning models and deep neural network algorithms. The feature extraction and the models used on the dataset helped to uniquely identify phishing URLs and also the performance accuracy of the models used. It is also surprisingly accurate at detecting the genuineness of a URL link.

Use anti-phishing protection and anti-spamsoftware to protect yourself when malicious messages slip through to your computer.

Anti-malware is included to prevent other types of threats. Similar to anti-spam software, anti-malware software is programmed by security researchers to spot even the stealthiest malware.

ACKNOWLEDGMENTS

We would like to express our special thanks to our guide D.Sreenivas Rao Sir who gave us a golden opportunity to do this wonderful project on this topic which also helped us in doing a lot of research and we came to know about so many new things. We are really thankful to them.
Secondly, We would also like to thank my friends who helped us a lot in finalizing this project within the limited time frame.