

Approach for Detection of Phishing Attacks

ICM2016006

Overview

Over the years there have been many attacks of Phishing and many people have lost huge sums of money by becoming a victim of phishing attacks. In a phishing attack emails are sent to users claiming to be a legitimate organization, where in the email asks the user to enter information like name, telephone, bank account number, important passwords etc. such emails direct the user to a website where the user enters these personal information. These websites also known as phishing websites now steal the entered user information and carry out illegal transactions thus causing harm to the user. Phishing website and their mails are sent to millions of users daily and thus are still a big concern for cyber security.

Abstract

It is a project of detecting phishing websites which are the main cause of cyber security attacks. It is done using Machine learning with Python

Characteristics of Phishing url

A phishing URL and the corresponding page have several features which can be differentiated from a malicious URL. For example; an attacker can register a long and confusing domain to hide the actual domain name (Cybersquatting, Typosquatting). In some cases attackers can use direct IP addresses instead of using the domain name. This type of event is out of our scope, but it can be used for the same purpose. Attackers can also use short domain names which are irrelevant to legitimate brand names and don't have any FreeUrl addition. But these types of web sites are also out of our scope, because they are more relevant to fraudulent domains instead of phishing domains.

Beside URL-Based Features, different kinds of features which are used in machine learning algorithms in the detection process are given below

1. URL-Based Features
 2. Domain-Based Features
 3. Page-Based Features
-

4. Content-Based Features

URL-Based Features

URL is the first thing to analyse a website to decide whether it is a phishing or not. As we mentioned before, URLs of phishing domains have some distinctive points. Features which are related to these points are obtained when the URL is processed. Some of URL-Based Features are given below.

- Digit count in the URL
- Total length of URL
- checking whether the URL is Typosquatting or not. (google.com → goggle.com)
- Checking whether it includes a legitimate brand name or not (apple-icloud-login.com)
- Number of subdomains in URL
- Is Top Level Domain (TLD) one of the commonly used one?

Domain-Based Features

The purpose of Phishing Domain Detection is detecting phishing domain names. Therefore, passive queries related to the domain name, which we want to classify as phishing or not, provide useful information to us. Some useful Domain-Based .Features are given below.

- Its domain name or its IP address in blacklists of well-known reputation services?
- How many days passed since the domain was registered?
- Is the registrant name hidden?

Page-Based Features

Page-Based Features are using information about pages which are calculated reputation ranking services. Some of these features give information about how reliable a website is. Some of Page-Based Features are given below.

- Global Pagerank
- Country Pagerank
- Position at the Alexa Top 1 Million Site

Content-Based Features

Obtaining these types of features requires active scan to target domain. Page contents are processed for us to detect whether target domain is used for phishing or not. Some processed information about pages are given below.

- Page Titles
- Meta Tags
- Hidden Text
- Text in the Body
- Images etc.

All of the features explained above are useful for phishing domain detection. In some cases, it may not be useful to use some of these, so there are some limitations for using these features. For example, it may not be logical to use some of the features such as Content-Based Features for the developing fast detection mechanism which is able to analyze the number of domains between 100.000 and 200.000. Another example would be, if we want to analyze new registered domains Page-Based Features is not very useful. Therefore, the features that will be used by the detection mechanism depends on the purpose of the detection mechanism. Which features to use in the detection mechanism should be selected carefully.

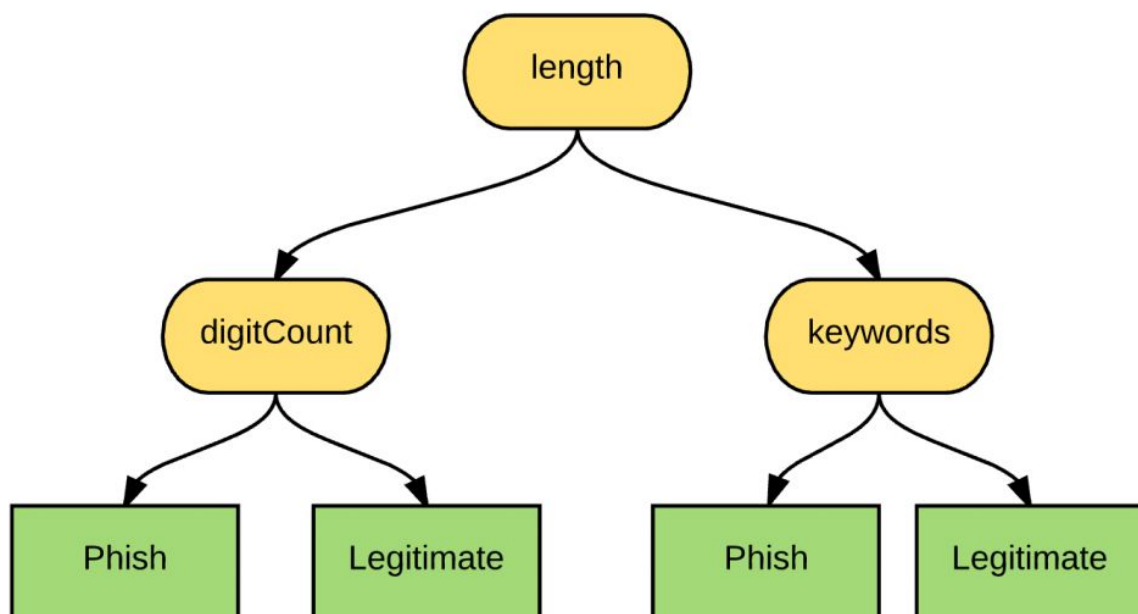
Detection Process

When we have raw data for phishing and legitimate sites, the next step should be processing these data and extract meaningful information from it to detect fraudulent domains. The dataset to be used for machine learning must actually consist of these features. The feature values should be selected according to our needs and purposes and should be calculated for every one of them.

phishing domain is one of the classification problems. So, this means we need labeled instances to build detection mechanisms. In this problem we have two classes: **(1)** phishing and **(2)** legitimate.

A Decision Tree can be considered as an improved nested-if-else structure. Each feature will be checked one by one.

Generating a tree is the main structure of the detection mechanism. Yellow and elliptical shaped ones represent features and these are called nodes. Green and angular ones represent classes and these are called leaves. The length is checked when an example arrives and then the other features are checked according to the result. When the journey of the samples is completed, the class that a sample belongs to will become clear.



Decision Tree uses an information gain measure which indicates how well a given feature separates the training examples according to their target classification. The name of the method is **Information Gain**. The mathematical equation of information gain method is given below.

$$Gain(S, A) = \underbrace{Entropy(S)}_{\text{original entropy of S}} - \underbrace{\sum_{v \in values(A)} \frac{|S_v|}{|S|} \cdot Entropy(S_v)}_{\text{relative entropy of S}}$$

High Gain score means that the feature has a high distinguishing ability. Because of this, the feature which has the maximum gain score is selected as the root. **Entropy** is a statistical measure from information theory that characterizes (im-)purity of an arbitrary collection S of examples. The mathematical equation of Entropy is given below.

$$H(S) \equiv \sum_{i=1}^n -p_i \log_2 p_i$$

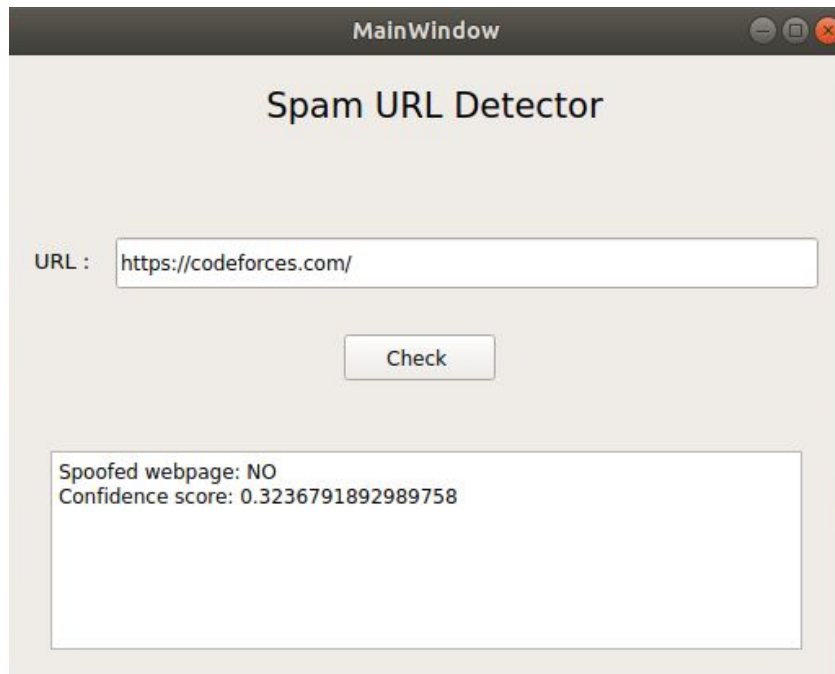
Original Entropy is a constant value, Relative Entropy is changeable. Low Relative Entropy Score means high purity, likewise high Relative Entropy Score means low purity. As we move down the tree, we want to increase the purity, because high purity on the leaf implies high success rate.

Accuracy of the Model

```
Training data loaded.  
Decision tree classifier created.  
Beginning model training.  
Model training completed.  
Predictions on testing data computed.  
The accuracy of your decision tree on testing data is: 0.906129210381
```

You can try improving the accuracy of this simple classifier by changing some of the default parameter values for the model

GUI of the Project



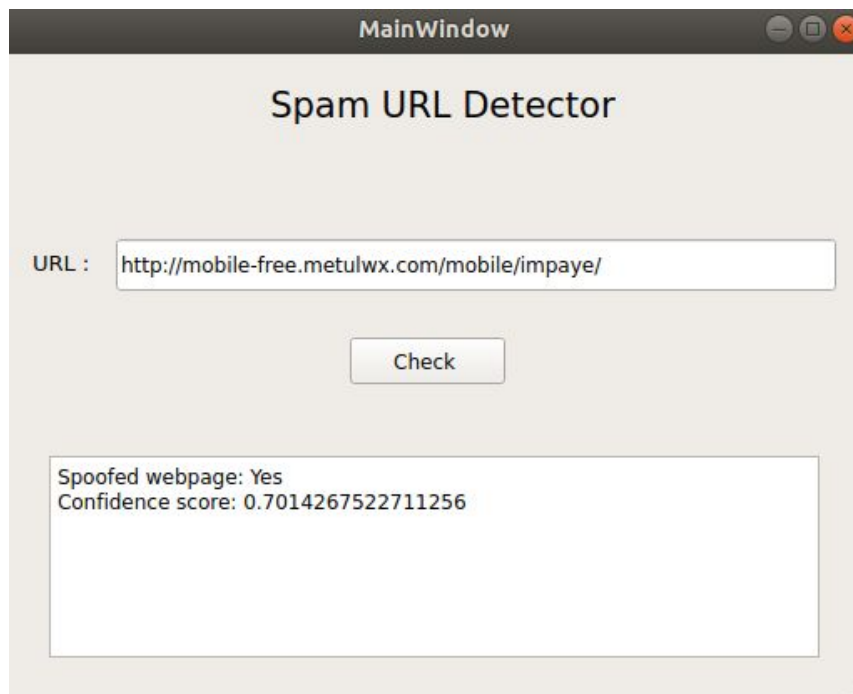
The screenshot shows a window titled "MainWindow" with the title bar containing standard OS controls. The main content area is titled "Spam URL Detector". Below the title, there is a text input field labeled "URL :" containing the text "https://codeforces.com/". Below the input field is a button labeled "Check". Below the button is a text area containing the results: "Spoofed webpage: NO" and "Confidence score: 0.3236791892989758".

MainWindow

Spam URL Detector

URL :

Spoofed webpage: NO
Confidence score: 0.3236791892989758



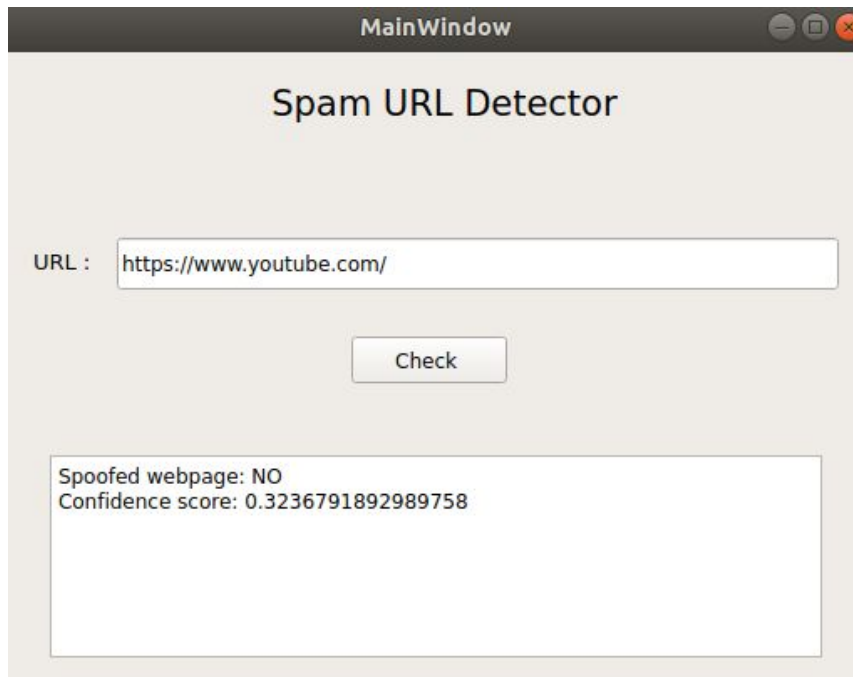
The screenshot shows a window titled "MainWindow" with the title bar containing standard OS controls. The main content area is titled "Spam URL Detector". Below the title, there is a text input field labeled "URL :" containing the text "http://mobile-free.metulwx.com/mobile/impaye/". Below the input field is a button labeled "Check". Below the button is a text area containing the results: "Spoofed webpage: Yes" and "Confidence score: 0.7014267522711256".

MainWindow

Spam URL Detector

URL :

Spoofed webpage: Yes
Confidence score: 0.7014267522711256



Future Scope

To implement a google extension using the javascript.

References:

- Basnet R., Mukkamala S., Sung A.H. (2008) Detection of Phishing Attacks: A Machine Learning Approach. In: Prasad B. (eds) Soft Computing Applications in Industry. Studies in Fuzziness and Soft Computing, vol 226. Springer, Berlin, Heidelberg