Final Project: Malicious Site Detector

Dan Saubert & Isaac Burmingham

21 August, 2020

## Data Source: Kaggle

* Source: <https://www.kaggle.com/vishalsiram50/phishing-websigtes-data>
* Column Explanation: “Phishing Website Features.docx”
* Type: csv, 31 Attributes per record

This data is a list of potential Phishing websites and their attributes. The attributes include information about the domain, website content, security protocols, and other factors that help determine if a website is more likely to be malicious. The attributes fall into a few categories, with each individual feature thoroughly described in Appendix A

* URL & Address Bar - Visual manipulation and redirection tricks that can be exploited from within the address bar of a web browser. This usually involved hiding characters inside the URL that the user might miss, or putting special character that the URL bar parses in an unexpected way.
* Webpage Content - Where external content is coming from and where the content you enter to the website is going. This includes images on the webiste and their source, as well as the webforms on a website and their destination.
* JavaScript & HTML - Interactive portions of the website that have been manipulated in a suspicious way. JavaScript is a powerful way to manipualte a website in realtime, which makes it a useful tool for phishers to load, move, and shift a website while the user interacts with it.
* Domain - Metadata about the domain gathered from external sources, like Google and PageRank. These metrics are mostly calcualted and reported by independent watchdogs that have a vested financial interest in keeping phishing websites off the internet.

## Research Question

Can malicious websites (phishing sites) be discovered using site attributes and metadata?

Phishing websites aim to trick users into handing over passwords, social security numbers, financial data, and other valuable private information. These sites, like all sites, have plenty of metadata about the site which can be analyzed. We aim to determine if phishing sites, in relation to legitimate websites, have statistically significant characteristics that can help identify these sites.

## Research Method

The overarching goal of our research question is classification. Random forests are an effective method for classifying things and have few data prerequisites to be utilized. We are using the built in R method for generating random forests, which takes care of both bagging (bootstrap aggregation) and feature randomness.

* Bagging: The process of allowing each individual tree to randomly sample from the dataset with replacement, resulting in different training sets for each tree.
* Feature Randomness: Each tree in the random forest can pick only from a random subset of features. This forces even more variation among the trees and results in lower correlation across trees.

In tandem, these two aspects ensure high tree diversity within the forest. Tree diversity is essential because the underlying theory that defines random forests states: “A large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models.”

The built-in R RandomForest uses a predictive metric to determine the error rate. While this is usually enough to find the conclusion satisfactory, we are going the extra step to produce a true test data set for analysis. In doing so, we can calculate a non-biased test statistic to prove that our Random Forest model is accurately predicting which websites are malicious or not.

## Data Satisfaction

As defined in our question, websites can fall into two categories: Legitimate and Phishing. Once a website uses malicious tactics in any way, it should be considered phishing, as no legitimate website would attempt to steal data in the way a phishing site would. This binary classification extends to the features in this data set. Some features are ternary, but most are {-1, 1}. Due to the discrete nature of the features, a binary decision tree is a great method for determining which features can separate the phishing sites from the legitimate ones. Furthermore, binary decision trees can be improved through the use of random forests, which are a collection of smaller decision trees that operate under the idea of crowd decision making.

The random forest only demands that the output be discrete (categorical), which our data certainly is.

## Method Application

Using the randomForest module, we ask the forest builder to find a correlation between the Result column (phishing or legitimate) and all other attributes in a record. By default, the module will automatically optomize the parameters for the features per tree and number of trees in the forest.

# Generate random forest model: -1 is a phising site, 1 is a legitimate website  
forest <- randomForest(Result ~ ., data = phishing\_data\_train, importance = TRUE)  
print(forest)

##   
## Call:  
## randomForest(formula = Result ~ ., data = phishing\_data\_train, importance = TRUE)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 5  
##   
## OOB estimate of error rate: 3.35%  
## Confusion matrix:  
## -1 1 class.error  
## -1 3745 166 0.04244439  
## 1 130 4803 0.02635313

**Forest error rate graph featured in Appendix B.**

## Method Accuracy Analysis

While the self reported OOB accuracy measure is likely sufficient, here we calculate a contingency table to determine the rate of false positives and negatives from the model. From this table, we can apply Fisher’s exact test to determine if the model is providing statistically significant website classifications.

# Calculate the 'true' accuracy of the model using the test data.  
resultsColumnNumber <- 31  
forest\_prediction <- predict(forest, phishing\_data\_test[, -resultsColumnNumber])  
accuracy <- table(observed = phishing\_data\_test[, resultsColumnNumber], predicted = forest\_prediction)  
# Accuracy contingency table, by row  
prop.table(accuracy, margin = 1)

## predicted  
## observed -1 1  
## -1 0.95947315 0.04052685  
## 1 0.02614379 0.97385621

# Fisher  
fisher.test(table(observed = phishing\_data\_test[, resultsColumnNumber], predicted = forest\_prediction))$p.value

## [1] 0

As observed in the fisher test above, the model is working extremely well (so well in fact that the p-value is small enough to be rounded to 0) and can be trusted as classifier of phishing websites. From this we can conclude that yes, phishing websites can be discovered using website metadata alone.

## Model Analysis & Conclusion

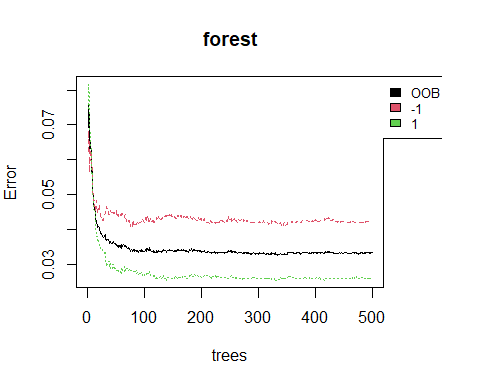
We’ve already proven our research question true by showing that we can predict phishing websites with statistical significance, but an additional feature of random forests gives us even more insight into those classifications through feature importance and node impurity. First, the feature importance is calculated by seeing which trees lose the most accuracy when features are omitted. Second is GINI, which is a measure of node ‘impurity’ which can be interpreted as which nodes have the most data from each class mixed together. If a node is particularly good at splitting the data into the correct categories, that node’s purity will be high. By polling each tree in the forest we can sort out which features are most important in successfully classifying the websites as legitimate or phishing.

Looking at the plot in Appendix C, its clear that SSL Final State, URL of Anchor, and Web Traffic are the top three most important features that can be used to spot a phishing website. SSL Final State is not a surprising find, as this attribute involves the most discretion from human experts and agencies. The only way to receive a non-suspicious score on this attribute is to certify your website through top domain hosting companies, which require detailed personal information about the the person making the website. Furthermore, this category is self-monitored and policed by those agencies, which means that once a site is reported malicious action is quickly and automatically taken to revoke that website’s certification.

## Appendix A

* URL & Address Bar - Data about the server that hosts the website and the address that it publicly broadcasts or tricks that can be exploited from within the address bar of a web browser.
  + Having IP Address
  + URL Length
  + Shortening Service - services that mask true URL identities like “bit.ly”
  + Abnormal URL - URL does not match core identity of domain.
  + Having @ Symbol
  + \\ Redirecting - Using a double slash can hide a URL within a URL and direct the browser to an unintended website automatically.
  + Prefix/Suffix - Uses a ‘-’ symbol in the URL
  + Sub-domain - URL’s that contain many ‘.’ symbols. The ‘.’ allows a website to mask multiple sub-websites within it.
  + SSL Final State - Final state of HTTPS protocol. In modern browsers this is quite important, as it is the firm handshake between your browser and the website, by which both agree that the other is trustworthy. A successful SSL state is difficult to spoof because there are entire companies and consortiums dedicated to ensuring only legitimate websites have the credentials and authority to reach this state.
  + Domain Registration Length - How long the websites domain is currently purchased for. Phishing sites are taken down frequently, so phishers rarely pay for more than a year or so, unlike legitimate websites which may pay for up to decades of registration.
  + Favicon - TODO: Picture of Favicon
  + Port - Internet protocol defines certain ports for different activities. Using anything besides 443 for HTTPS is a sign of potential malice.
  + HTTPS Token - Using the string ‘https’ inside the URL to trick the user into thinking the site is secure.
* Webpage Content - Where external content is coming from and where the content you enter to the website is going.
  + Request URL - Examines the content within the webpage and determines how much of the content is loaded from the same domain the the host URL.
  + URL of Anchor - Anchors are links to multiple types of content, whether it be images, buttons, or any other content externally loaded. These URLs are hidden from users and are a difficult to catch by human eye.
  + Links in Tags - Links hidden in parts of the webpage that should not contain them.
  + Server Form Handler (SFH) - An SFH handles information entered by the user. On a malicious site, this handler is usually pointed to a different domain to steal information from the user.
  + Submitting to Email - Information submitted to the webpage is sent to an email, instead of a server, which only occurs on phishing sites.
* JavaScript & HTML - Interactive portions of the website that have been manipulated in a suspicious way.
  + Redirect - Redirecting a webpage is normal during a login, but too many redirects are quite common for phishing sites.
  + On Mouseover (URL Spoofing) - JavaScript contains a function that will detect if the users mouse is hovering over an element. For phishing websites, they will put this function over the entire webpage so that it’s always true. In doing so, they can have that event trigger an update of the URL of the webpage without actually navigating anywhere.
  + Right Click - Blocking the ability to right click hinders the ability to see the webpage source.
  + Pop Up Window - Pop up windows that ask for user information are extremely rare on legitimate websites.
  + iFrame - iFrame allows an entire second webpage to be shown within a webpage. Not only this, but the phishing site can be entirely invisible, operating invisibly on top of a legitimate website.
* Domain - Metadata about the domain gathered from external sources, like Google and PageRank.
  + Age of Domain
  + DNS Record - When a website is registered, a WHOIS profile is built around the company or person who did so. Phishers will commonly use fake information or no information at all in this area.
  + Web Traffic - Number of visitors as reported by Alexa (the Web Information Company)
  + Page Rank - A rank of page “importance”, most phishing sites have a rank of 0 (no importance)
  + Google Index - Google’s web crawler is extremely good at discovering and indexing websites. Phishing links are rarely placed on websites and most sites live for only a few months, making them unlikely to be found in Google’s index.
  + Links Pointing to Page
  + Statistical Report - Companies offer services that track and expose top phishing domains. Their efforts are certainly reminiscent of “Whack-a-Mole”, but they prevent phishers from getting lazy are reusing old tactics.

## Appendix B



## Appendix C

A screenshot of a computer

Description automatically generated