# **NSF RET Research Project 2019**

**Team C Project #1**

# **Title: Phishing Detection using URL Attributes**

**Project Overview:**

One of the biggest security threats on the web today is phishing. Through phishing attacks, the victim might be lured into several traps, such as clicking on a link to a malicious website, entering private information into a form controlled by the adversary, or even directly installing malware on the end-host. To combat this attack vector, phishing detection is now a key attribute of many security solutions.

In this project, you will gain insights into how some of these security solutions flag potential phishing links. You will work with a large dataset containing a mix of phishing links and non-phishing links and train machine learning classifiers to discriminate between these two kinds of links. The dataset was collected from 2456 websites among which some were used for phishing and others not. For each website included in the dataset, 30 features were extracted. In the paper entitled: *An Assessment of Features Related to Phishing Websites using an Automated Technique*, you will find a detailed description of the data and its features.

Using this dataset you will explore the full data analytics pipeline, including steps such as, data loading, cleaning, feature transformation and dimensionality reduction, classifier training and model evaluation. Figure 1 provides an overview of the steps that you will undertake using R.

Data Set Loading/ Analysis

Real Time Data / New data

**Machine Learning Model**

Best Model

Data Cleaning

Result / Decision

Further data processing as per algorithm needs (Scale/train/test)

Model performance Evaluation

Figure 1: Overview of Data Analytics Pipeline

**Project Steps:**

1. **Data Loading:**

The very first step of a data analytics project is to load data into the programming environment. Being at the start of the data analytics pipeline, any errors committed at this step might affect the rest of the process. For example, specific attention has to be paid to the *stringsAsFactors* option used by some of the data loading functions. If this option is not set right, some variables might load in a form that is different from what certain algorithms expect. If this issue is not addressed in the code, some algorithms might abort or simply produce wrong results.

Another noteworthy issue about data input in R is that certain functions are much faster than others, and thus a better option for larger datasets. For data analytics tasks requiring repeated loading of data, a poor choice of data loading function might tremendously slow down the data analytics pipeline. In this part of the project, you will explore some of the data loading functions, compare their performance and study the impact of changes in some of their parameter settings.

You will use two datasets, a small dataset (in the file rawDataSetSmall.csv) and a large dataset (in the file rawDataSetLarge.csv). You might need the following libraries: ff, bigmemory, sqldf and data.table. Remember to install the corresponding packages before importing these libraries.

Please follow the following steps to complete this task.

* 1. Use the function *read.csv* to read the smaller of the two datasets above in two ways: (1) with the *stringsAsFactors* set as false*, and (2)* with this option not set*.* In both cases use the str() function to examine the data loaded in the R environment. What differences do you observe between scenarios (1) and (2)?
  2. For each of the smaller and larger datasets, use the following functions to load the data into a variable.
     1. read.csv()
     2. fread()
     3. read.csv.ffdf()

Estimate how long it takes each of the functions to read the data. For example, you can use:

Timing=*system.time*(*read.csv*("rawDataSetSmall.csv")) to determine how long it takes the function *read.csv* to read the rawDataSetSmall.csv dataset.

Create a table to summarize the loading time for each function and dataset.

Which of these functions would you choose for a big dataset?

1. **Descriptive Analysis:**

In this segment of the project you will undertake an exploration of the structure of the data and read the associated paper to make sense of the features used by the phishing classification engine.

* 1. Load the small dataset and use the str command to find out the header information of features.
  2. Describe *at least six* of the features listed below and the basic intuition behind why they might help detect a phishing url (This task will require you to read the paper entitled: *An Assessment of Features Related to Phishing Websites using an Automated Technique*).
     1. having\_IP\_Address
     2. URL\_Length
     3. Shortining\_Service
     4. having\_At\_Symbol
     5. double\_slash\_redirecting
     6. Prefix\_Suffix
     7. having\_Sub\_Domain
     8. SSLfinal\_State
     9. Domain\_registeration\_length
     10. Favicon
     11. Port
     12. HTTPS\_token
     13. Request\_URL
     14. URL\_of\_Anchor
     15. Links\_in\_tags
     16. SFH
     17. Submitting\_to\_email
     18. Abnormal\_URL
     19. Redirect
     20. on\_mouseover
     21. RightClick
     22. popUpWidnow
     23. Iframe
     24. age\_of\_domain
     25. DNSRecord
     26. web\_traffic
     27. Page\_Rank
     28. Google\_Index
     29. Links\_pointing\_to\_page
     30. Statistical\_report
     31. Result

Notice that the features have been coded to have the values 1,-1; or -1,1,0. Using the description provided in the paper for this data for guidance, explain the meaning of this coding.

1. **Data Cleaning:**

Data scientists spend the majority of their time (some estimates put this at 80% of their time) cleaning and manipulating data and only a very small proportion (say, 20%) of their time actually analyzing it. For this reason, it is critical to become familiar with the data cleaning process and all of the tools available to you along the way. One common data cleaning operation is to handle missing values, denoted NA in R. Sometimes we may want to ignore any missing values, but other times we may wish to impute or estimate the missing values.

* 1. Using summary command or *str()* in R, find out if the data in the file, *rawDataSet.csv*, contains any missing values. If yes,
     1. How many missing values does it have?
     2. Can we proceed with our project without cleaning the missing values? What are possible problems if we do not replace missing values?
  2. In this section you will perform two methods of missing value imputation and visualize the data to verify the imputation.
     1. Imputing missing values with the mode: Copy the code from the file *modeImputation.R* into your environment and run this code. This code has a function that replaces missing values in each column with the mode. You will call this function as follows: *modeImputation* (*dataIn*), where *dataIn* is the variable holding the data that you read from the file: *rawDataSet.csv.* Save your result in the variable: *cleanDataSet*
     2. Handling missing values by deleting their associated data instances: Use the function *na.omit ()* to delete instances with missing values. Save your result in the variable: OmitMissingValues.
     3. For a visual verification of the operations performed on the missing values, use a plot of your choice on the original dataset, the cleanDataset and the one with missing values. Discuss your observations.

1. **Data Visualization:**

Data visualization is one of the most powerful tools for understanding the properties underlying a dataset. Often the decision to perform certain operations on a dataset begins with a visualization step whose findings guide the researcher on what steps to undertake to make the inferences required for the data. In this section you will perform some basic visualization tasks on the dataset (*from this section onwards, all your work will be based on the clean dataset*). Some of the plots that you will create will require you to do some research online to find the relevant R functions.

* 1. Plot the data using a correlation matrix. Also plot a heat map to visually capture the correlations. Based on this matrix/heat map, discuss what you learn about this data.
  2. Plot the data using three different types of plots and discuss what the plots say about this data
     1. Box plot
     2. Histogram of class labels
     3. Histogram of two variables of your choice.

1. **Model Building, Prediction and Model Evaluation:**

The final steps in most data analytics projects are to build (or train) some model, perform predictions using this model, and evaluate the performance of this model. Often this process is iterative, going on with variations in parameters until the best model can be found. You will apply the following data modeling algorithms in your research: Neural Networks (NN) and K-Nearest Neighbors (K-NN).

* 1. Using the caret package, split the *cleanDataSet* into a training set and a test set. Use *p=0.75* (i.e., 75% of the data is in the training set).
  2. Operations with K-NN:
     1. Using the training set, train a K-NN model for each of K=2, K=6, K=10.
     2. Using the test set, perform predictions with each of the trained K-NN models.
     3. Create a crosstable for each model and discuss the results. Which model performs best? Why?
  3. Operations with NN:
     1. Using the training set, train a NN for each of n=2, 3 and 4, where n is the number of nodes in the hidden layer.
     2. Plot the neural network trained in each of these models (recall visualization function in the neuralnet package used in class).
     3. Using the test set, perform predictions with each of the trained NN models
     4. Create a crosstable for each model and discuss the results. Which model performs best? Why?
  4. Repeat steps (a) through (c) using p= 0.5 and p=0.3.

1. **Conclusion**
   1. Discuss the results obtained above and draw your conclusions. Your discussion should include a comparison of the performance of the 2 classification algorithms as well as the impact of the various parameters used (e.g., K, p, n).

**ACKNOWLEDGMENT:**

Graduate students “Mehta Santoshkumar” and “Bilal Nasir” prepared the first version of this project.