**Malicious URL Identification**

Mark Gerken

Dr. Kevin Scannell

Machine Learning – CSCI 4930

**Project Description**

This project strove to achieve the highest accuracy possible in the identification of malicious URLs. The dataset used for training and testing was retrieved from <https://www.sysnet.ucsd.edu/projects/url/>. The URLs are contained in an uncompressed archive of an anonymized 121-day dataset consisting of a total of about 2.4 million URLs and up to 3.2 million features.

**Data Exploration**

The data archive contains 121 independent files. Each is a .svm file corresponding to one day of the dataset, and labeled DayX.svm (where X is an integer from 0 to 120). URLs are labeled +1 if they are malicious, and -1 if they are benign. Each SVMlight file consists of a Scipy sparse matrix for the data and a Numpy ndarray containing the corresponding labels for each data vector. The features are reported to correspond to different values such as if the URL has been reported on blacklists, properties gathered from the associated IP address, WHOIS information, domain name, and even geographic properties.

I reviewed three different research papers (appropriately sourced at the end of this report) to see what I could learn about the data set features. Unfortunately, while all three papers mention generally mention the features, none give specifics on each features exact meaning. I explored the possibility of trying to weight certain features higher than others, but with over 3.2 million features, it became quickly apparent this would not give any worthwhile results.

**Evaluation Process**

The dataset was immediately segregated for training and testing. The first 90 days of the data was taken for training. The files corresponding with days 90-120 were moved to a different location to be used exclusively for testing. Most statements of accuracy in this report should be taken as a close estimate for the performance, unless explicitly stated otherwise. Due to the sheer size of the working dataset, I was unable to run full tests (i.e. training all 90 days, and testing all 31 days) every single time I made a change to classifiers as the full test would take several hours to complete depending on the classifier. Instead, training and testing was done by taking stochastic subsets of the data. I selected and reduced the first five files of both the training and test data to only include the first 3000 data points, which I will refer to as my *sample* files. This reduction drastically reduced runtime, and as such were the files used for minor tweaks and experimentation. Periodic larger tests were also performed in which two different arrays were randomly generated with a uniform distribution: one four integer array to select training files and a two integer array to select testing files. These arrays were used to select which (full-sized) files would be used. This randomness allowed me to test various files over many trials with minimal favoritism or bias.

**Process and Results**

SVM Classifier

The first objective was to establish a bare minimal baseline of classifier. I first tested used the sample files to test the different base SVM classifiers with default parameters.

Training Data: (Sample) Day0, Day1, Day2

Testing Data: (Sample) Day90

RBF: 68.22%

Linear: 97.03%

Poly (5 degree): 69.23%

Sigmoid: 69.23%

From these results, pursuing a linear classifier further was obviously the best choice. I wanted to get a better breakdown of where the linear classifier was making its mistakes, so I reran the classifier training against the full Day0, Day1, and Day2 file. I then tested against the full Day90 file, and created a confusion matrix based on those predictions.

Accuracy: 97.86%

|  |  |  |
| --- | --- | --- |
|  | TRUE | FALSE |
| POSITIVE | 13322 | 218 |
| NEGATIVE | 210 | 6250 |

KNN

I then experimented with a KNN implementation. Stand alone, the KNN classifier gave the following results.

Training (Sample): Day0, Day1, Day2

Testing (Sample): Day 90

Accuracy: 90.73%

|  |  |  |
| --- | --- | --- |
|  | TRUE | FALSE |
| POSITIVE | 2005 | 72 |
| NEGATIVE | 206 | 717 |

Due to the significantly higher portion of True Negatives, I then attempted to do an aggregation of the linear classifier