Learning to Detect Malicious URLs

Kaveri A, Lasya V, Sreekavitha P

Group 44

International Institute of Information Technology, Hyderabad

***Abstract*—Malicious Web sites are a cornerstone of Internet criminal activities. The dangers of these sites have created a demand for safeguards that protect end-users from visiting them. This paper explores how to detect malicious Web sites from the lexical and host-based features of their URLs. URL reputation as a binary classification problem where positive examples are malicious URLs and negative examples are benign URLs.**

I. Introduction

For almost as long as the commercial world wide web has existed, annoying advertisements have competed with desired content for users' attention. To combat these annoyances,users have created methods for filtering out the unwanted ads. While the problem of identifying privacy-violating advertise- ments has received a large amount of attention from the research community (as outlined in Section 2), the current state of the art for end user ad blocking remains a process with several tedious manual steps.

II. Application

A. Problem Overview

Uniform Resource Locators (URLs), sometimes known as "Web links," are the primary means by which users locate resources on the Internet. Our goal is to detect malicious Web sites from the lexical and host-based features of their URLs.

URL reputation as a binary classification problem where positive examples are malicious URLs and negative examples are benign URLs. We classify sites based only on the relationship between URLs and the lexical and host-based features that characterize them.

B. URL Resolutions

URLs have the following standard syntax. <protocol>://<hostname><path>

<protocol> Indicates which network protocol should be used to fetch the requested resource.

Benign URL: Has HTTP,HTTPS,FTP

MaliciousURL:http://malicioussite.com/http:/www.cs.ucsd.edu.

<path> of a URL is analogous to the path name of a file on a local computer.

Benign URL: The path tokens, delimited by various punctuation such as slashes, dots, and dashes, show how the site is organized.

Malicious URL: Criminals sometimes obscure path tokens to avoid scrutiny, or they may deliberately construct tokens to mimic the appearance of a legitimate site, as in the case of phishing.

<hostname> The is the identifier for the Web server on the Internet.

Malicious URLs:

The Domain Name System (DNS) is a hierarchical network of servers responsible for translating domain names into IP addresses and other kinds of information.

The A, MX, and NS records are IP addresses of hosts associated with a domain name. Under the hypothesis that the hosting infrastructure for malicious URLs is distinct from that for benign URLs, the various DNS records become useful differentiators. The A records for malicious Web servers might be hosted on compromised residential machines or in disreputable providers.

Benign URLs:

there is a special DNS record type called the pointer (PTR) record. Its purpose is to enable reverse DNS lookups: given an IP address as a query, the associated domain name is returned.

The existence of a PTR record is a reliable indicator that the domain name is well established.

Besides the IP addresses associated with a domain name, there is useful information associated with domain name registration. This information includes vital data about the registrant, the registrar, date of registration, date of expiration, date of the latest update, and other attributes associated with the record.

C. Features

For the classifier to be effective, we attempted to identify features that may differentiate an ad-related URL from a non ad-related URL. These include characteristics of the structure of the URL, keywords present in the URL, the container it was requested from on a page, and other page properties. All features are either binary or real-valued between 0 and 1. Conveniently, this approach allows all features to be given equal weight at training time. Below is the list:

Having_IP_Address { -1,1 }
URL_Length { 1,0,-1 }
Shortining_Service { 1,-1 }
Having_At_Symbol { 1,-1 }
Double_slash_redirecting { -1,1 }
Prefix_Suffix { -1,1 }
Having_Sub_Domain { -1,0,1 }
SSLfinal_State { -1,1,0 }
Domain_registeration_length { -1,1 }
Favicon { 1,-1 } Port { 1,-1 }
HTTPS_token { -1,1 }
Request_URL { 1,-1 }
URL_of_Anchor { -1,0,1 } Links_in_tags { 1,-1,0 }

Having_IP_Address { -1,1 }
URL_Length { 1,0,-1 }
Shortining_Service { 1,-1 }
Having_At_Symbol { 1,-1 }
Double_slash_redirecting { -1,1 }
Prefix_Suffix { -1,1 }
Having_Sub_Domain { -1,0,1 }
SSLfinal_State { -1,1,0 }
Domain_registeration_length { -1,1 }
Favicon { 1,-1 } Port { 1,-1 }
HTTPS_token { -1,1 }
Request_URL { 1,-1 }
URL_of_Anchor { -1,0,1 } Links_in_tags { 1,-1,0 }

Logically as deduced by us these features can be broadly classified as the following:

1. Address bar features
2. Abnormal features
3. HTML JS features
4. Domain features

The feature vectors for this paper have been provided at the following URL: https://archive.ics.uci.edu/ml/datasets/URL+Reputation

D. Classification Models

Perceptron

In machine learning, the perceptron is an algorithm for supervised learning of binary classifiers: functions that can decide whether an input belongs to one class or another.

SVM [RBF, Polynomial, Linear, Sigmoid]

*I*n machine learning, the (Gaussian) radial basis function kernel, or RBF kernel, is a popular kernel function used in various kernelized learning algorithms. In particular, it is commonly used in support vector machine classification.

In machine learning, the polynomial kernel is a kernel function commonly used with support vector machines (SVMs) and other kernelized models, that represents the similarity of vectors (training samples) in a feature space over polynomials of the original variables, allowing learning of non-linear models.

Logistic Regression

Logistic regression can be seen as a special case of generalized linear model and thus analogous to linear regression. Logistic Regression is a type of probabilistic classification model that is modeled by the re- lationship between a dependent variable and one or more independent variables. The decision for a point's label is determined by it's distance from a hyperplane decision bound- ary that is estimated at training time.

PassiveAgressive

PA is online classification algorithm.

Naive Bayes

In machine learning, naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features.

Decision Trees

Decision tree learning uses a decision tree as a predictive model which maps observations about an item to conclusions about the item's target value. It is one of the predictive modelling approaches used in statistics, data mining and machine learning. Tree models where the target variable can take a finite set of values are called classification trees.

KNN

In pattern recognition, the k-Nearest Neighbors algorithm (or k-NN for short) is a non-parametric method used for classification and regression.[1] In both cases, the input consists of the k closest training examples in the feature space. The output depends on whether k-NN is used for classification or regression.

Adaboost

AdaBoost*, short for "Adaptive Boosting", is a machine learning meta-algorithm. It can be used in conjunction with many other types of learning algorithms to improve their performance. The output of the other learning algorithms ('weak learners') is combined into a weighted sum that represents the final output of the boosted classifier. AdaBoost is adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers. AdaBoost is sensitive to noisy data and outliers. In some problems, however, it can be less susceptible to the overfitting problem than other learning algorithms.*

III. ANALYSIS

Classification accuracy suffered maximum dip when Address bar features were removed. This was followed by the classification with Abnormal feature removal. This was consistent across all classifiers. HTML and Javascript features and Domain based features followed next with both their contributions varying across different classifiers.

We have conducted a fold by fold analysis of the data, a few of those graphs with their average accuracies by removing a particular feature for folds 2,3,4,5.

The performance for *SVM RBF and Polynomial kernels* are comparable (92.18605 and 92.40316). This shows a underlying non-linear behaviour of the data. This is further exemplified by a poorer performance by the Linear SVM and the Sigmoid SVM.

Perceptron and *Passive Aggressive* have similar performances(86.448377 and 88.791333). This can be attributed to the fact that Passive Aggressive also changes weights in a way similar to the gradient descent in Perceptron (it just penalize more aggressively i.e. performs greater gradient descent when the error is larger).

Gaussian Naive Bayes gives poorest accuracy (65.112043). This shows that the Gaussian distribution model may not be as well suited to replicate the probability distribution of the given dataset. Bernoulli distribution performs slightly and so it can be concluded that this distribution model is better in replicating the probability distribution of the given dataset. However, better performance is delivered by other classifiers.

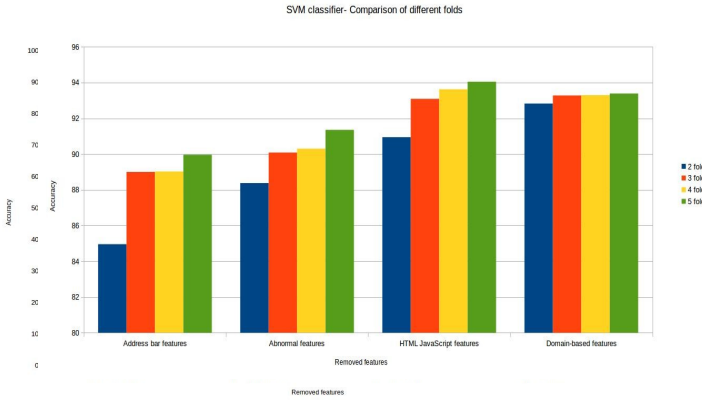
Toying around with the various combinations of features reveals that Gaussian performs the best(82.74) when Address bar features are removed where the rest of the classifiers perform poorly (opposite behavior), but Bernoulli performs the poorest(74.90) with these features removed. Gaussian performs poorest with the Abnormal features removed(57.55) and Bernoulli the best(90.12) with the HTML and Javascript features removed. So, we can say that Gaussian models the probability distribution the best when Address Bar features are removed and Bernoulli when the HTML and Javascript features.

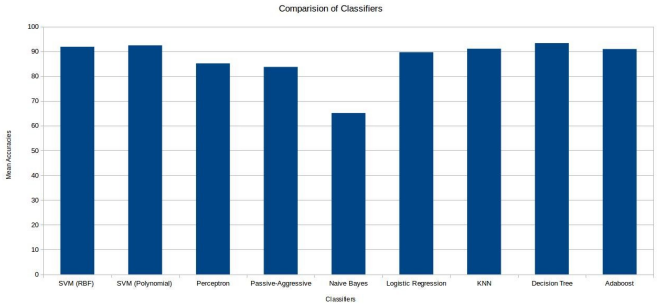
Decision tree is giving the highest accuracy (93.816575). This could be due to the fact that the features are completely binary (not real valued) and a tree can be made easily with forming rules with the given 31 features (which is a comparatively small number) i.e. lesser the number of features, more rules can be formed which can effectively classify the dataset (try comparing using 10 rules with a 2 vs. a 3 dimensional feature vector dataset).

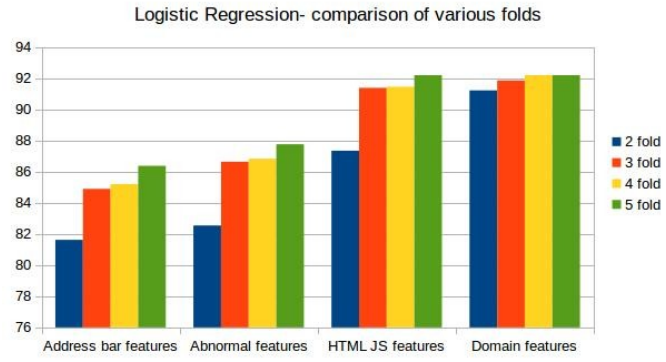
Logistic Regression and *KNN* have similar performances (89.630372408 and 91.061781794) among the top performing classifiers. Logistic Regression can be performing well again due to the no-linear distribution of data.

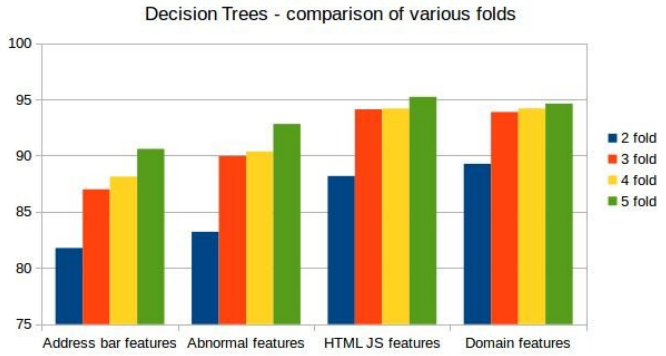
Adaboost's performance is just trailing behind the top 2 classifiers (SVM-RBF and Decision Trees) because it is also taking into consideration, the classification by the other underperforming classifiers.

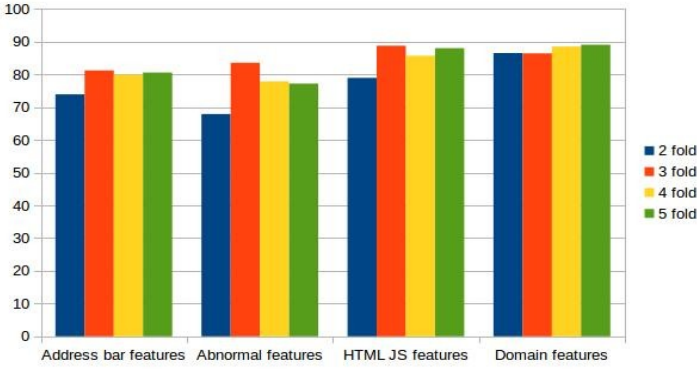
Figures and Tables











CONCLUSION

Classification accuracy suffered maximum dip when Address bar features were removed. This was followed by the classification with Abnormal feature removal. This was consistent across all classifiers. HTML and Javascript features and Domain based features followed next with both their contributions varying across different classifiers.

Hence Address bar features were the most informative and HTML and Javascript and Domain-based features were the least informative features to distinguish between benign and malicious URLs.

REFERENCES

[1] Learning to Detect Malicious URLs
JUSTIN MA, University of California, Berkeley, LAWRENCE K. SAUL, STEFAN SAVAGE and GEOFFREY M. VOELKER, University of California, San Diego.
http://cseweb.ucsd.edu/~savage/papers/TIST11.pdf

[2] Leveraging Machine Learning to Improve Unwanted Resource Filtering
Sruti Bhagavatula∗ Christopher Dunn† Chris Kanich∗ Minaxi Gupta† Brian Ziebart∗
ttps://www.cs.uic.edu/~ckanich/papers/bhagavathula2015leveraging.pdf