# Learning based Malicious Web Sites Detection Using Suspicious URLs

Haotian Liu, Xiang Pan, Zhengyang Qu *{haotianliu2011, xiangpan2011,zhengyangqu2017}@u.northwestern.edu*

EECS 349: Machine Learning, Bryan Pardo (instructor), Northwestern University

## 1. Motivation

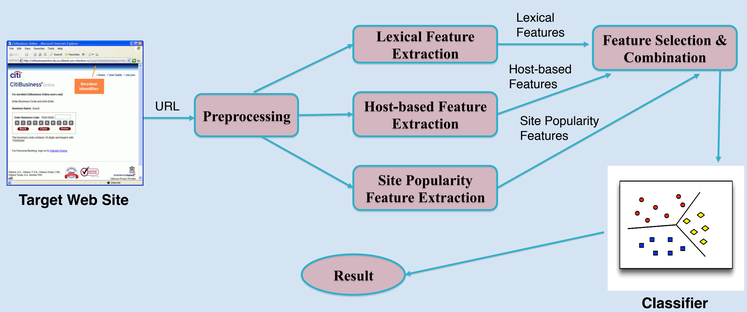
### Problem statement

Web applications become popular and bring people convenience, while there is a rapid growth in the number of attacks from various criminal enterprises, such as financial fraud and spam-advertised commerce. The common thread among those attacks is the requirement that unsuspecting users visiting the sites, clicking the target Uniform Resource Locator (URL). If we can be informed of the properties of the target URL in advance, in other words, whether it is dangerous or not, such problems will be largely resolved.

### Project Goal

This project is to present a learning based system that is capable of classifying URL intelligently without accessing the Web site, which removes the run-time latency and protects user from being exposed to browser-based vulnerabilities. Thus, our approach is complementary to existing blacklisting service and the systems based on evaluating site content and behavior. With the effective extraction of URL features and the resulting classification model, our system labels whether the Web site of the input URL is *benign*, *phishing*, or *malware*.

## 2. System




### Gathering The Data

We randomly collect 22,190 benign URLs from DMOZ Open Directory Project. DMOZ is one of the largest human-edited directory of the world. It classifies URLs into different categories. As for phishing URLs, we collect 5,703 samples from PhishTank, a collaborative site where people can submit and verify phishing URLs. Besides, we select 9,220 URLs from DNS-BH project, a site creating and maintaining a list of domains known to be used to propagate malware and spyware.

### Training

URLs in the dataset are labeled. We use 5-fold method to train our systems. Before training, we preprocess the features not consistent with others. For example, the range of traffic feature is much larger than that of other features. We map the feature into a much smaller range and it turns out to significantly increase the accuracy. We also use Chi-Square test and virtualization tool in Weka to select most informative features. Then, we use three learning algorithms-J48 Decision Tree, Support Vector Machine (SVM) and Logistic Regression to construct the model.

## 3. Approach

### Feature Extraction & Selection

We separate the features of URL into 3 classes: *Lexical Feature*, *Site popularity Feature*, and *Host-based Feature.*

Lexical Feature

| Feature | Type |
|---|---|
| *Length of hostname* | *Integer* |
| *Length of entire URL* | *Integer* |
| *Number of dots* | *Integer* |
| *Top-level domain* | *Integer* |
| *Domain token count* | *Integer* |
| *Path token count* | *Integer* |
| *Average domain token length* | *Real* |
| *Average path token length* | *Real* |
| *Longest domain token length* | *Integer* |
| *Longest path token length* | *Integer* |
| *Brand name presence* | *Binary* |
| *IP address presence* | *Binary* |
| *Security sensitive word presence* | *Binary* |

Host-based Feature

| Feature | Type |
|---|---|
| *Autonomous system number* | *Integer* |
| *IP country* | *Integer* |
| *Number of registration information* | *Integer* |
| *Number of resolved IPs* | *Integer* |
| *Domain contains valid PTR record* | *Binary* |
| *Redirect to new site* | *Binary* |
| *All IPs are consistent* | *Binary* |

Site Popularity Feature

| Feature | Type |
|---|---|
| *Number of External Links* | *Integer* |
| *Real traffic rank* | *Integer* |
| *Domain in a reputable sites list* | *Binary* |

### Method

Given the input URL, our system first extracts the features in 3 groups and performs transformation on the features with String type by referring to a Map constructed by us. With the sample and classification model, our approach can classify the URL and output the classification result.

### Learning Algorithm

J48 Decision Tree (DT)

Support Vector Machine (SVM)

Logistic Regression (LR)

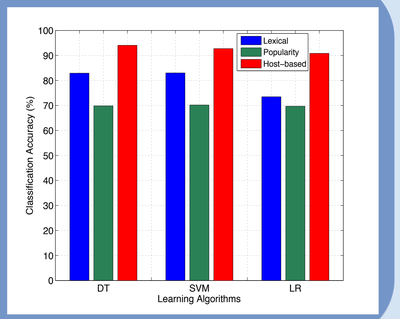
## 4. Evaluation

### Baseline

We adopt ZeroR algorithm as baseline method. ZeroR is a simple rule-based classifier and it assigns every value to the most common class. In our dataset, we have 37113 instances, including 5703 phishing URLs, 22190 benign URLs and 9220 malware URLs. Therefore, ZeroR will simply classify all the instances as benign URLs and in our case, the accuracy of ZeroR can be calculated by using the number of benign URLs divided by the size of dataset. The baseline is 59.79%.

### Feature Comparison

- Evaluate how much each group of features increases/decreases the classification accuracy
- Host-based features contribute the most to the performance.
- The effect of lexical features on classification accuracy will be largely reduced when LR is utilized.

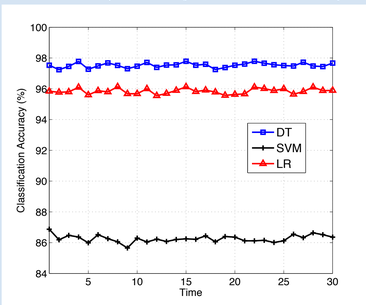



### Classifier Comparison

We evaluate each algorithm in classification accuracy, model building time, and true positive rate (TPR) and false positive rate (FPR).

| Classifier | Accuracy (%) | Runtime (s) |
|---|---|---|
| DT | 97.53 | 1.9 |
| SVM | 86.26 | 579.4 |
| LR | 95.84 | 5.7 |

30 times test (randomly select 75% samples)




DT

| | Benign | Malware | Phishing |
|---|---|---|---|
| TPR (%) | 98.3 | 99.9 | 90.7 |
| FPR (%) | 3.6 | 0.2 | 1.1 |

SVM

| | Benign | Malware | Phishing |
|---|---|---|---|
| TPR (%) | 98.9 | 84.8 | 44.0 |
| FPR (%) | 30.6 | 0.7 | 0.3 |

LR

| | Benign | Malware | Phishing |
|---|---|---|---|
| TPR (%) | 97.5 | 100 | 83.2 |
| FPR (%) | 6.4 | 0 | 1.8 |

## 5. Conclusions

We propose a learning based approach to separating Web sites into 3 classes: benign, phishing, and malware. The analysis is only based on URL itself without accessing the target website, which removes the run-time latency and protects user from being exposed to browser-based vulnerabilities. We argue that this approach is complementary to both blacklisting and the systems based on evaluating site content and behavior. By carefully selecting features and learning algorithms, our system achieves 97.53% accuracy on detecting malicious Web sites.

## 6. Web Site

http://users.eecs.northwestern.edu/~hlc720/349/index.html

Additional thanks to Zhiyao Duan, Mark Cartwright