CONCORDIA UNIVERSITY
Department of Computer Science and Software Engineering
COMP 6321 – Machine Learning
Summer 2018

Project Report
On
Analysing Suspicious URLs using Machine Learning Algorithms

By
Mandeep Kaur
Student ID – 40059801
k_ndeep@encs.concordia.ca

INTRODUCTION

Suspicious URLs are serious threat to cyber security and are more common these days. A user can be tricked by voluntarily giving away his/her personal information on a malicious webpage or can become a victim of downloads resulting in virus/malware infection. Hence, user must decide every time whether to click on any anonymous website without analysing risk. The project illustrates Machine Learning Algorithm to identify Malicious URLs. Additionally, document reflects detail results and possible future scope to improve the scope of the project. The aim of this project is to detect fishing websites by doing analysis of existing datasets containing suspicious URLs using well known Machine learning algorithms which are Logistic Regression and Support Vector Machine. This will mainly focus on extracting lexical and host-based features of URLs for determining their suspicious characteristics.

MOTIVATION

The main interest to do any machine learning project is to learn an algorithm totally based upon an application. While reading various machine learning projects from the list of research paper mentioned on the course website, Identification of malicious URLs found so unique and such a real world problem that we should be concerned about . Everyday whenever we browse something on Search Engine, suspicious URLs are the most common threat faced on Internet. This interested me to go for this project.

Basic Approach

This section represents the flow of the project model implemented and detail explanation of algorithms used.

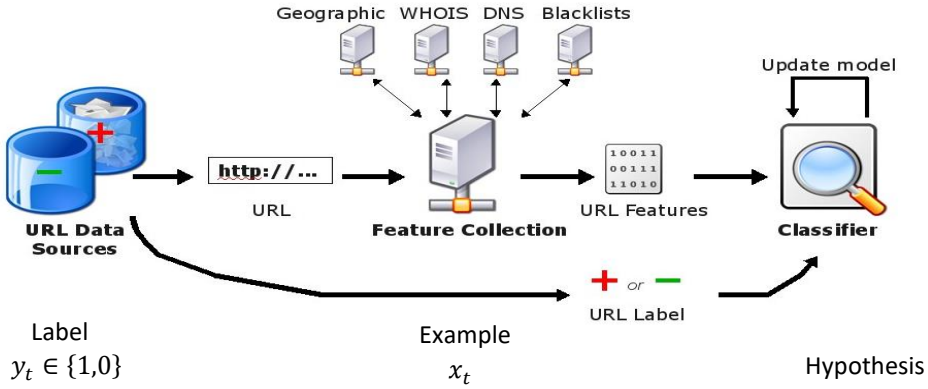


Figure 1 URL Classification Model

The Figure illustrates the overview of the system which has three main components. First component consists Malicious and benign URLs training data with labels 1 and 0 respectively. Second component represents a system that collects various features of the urls such as lexical and host based(explains in detail in upcoming section).Last component is the classifier which is used to classify URL features to identify whether the url is malicious or not based on the prediction.

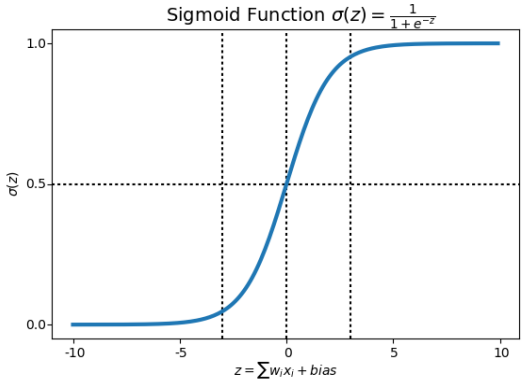
CLASSIFICATION ALGORITHMS

LOGISTIC REGRESSION

Logistic regression measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function, which is the cumulative logistic distribution. The logistic regression model is simply a non-linear transformation of the linear regression. The "logistic" distribution is an S-shaped distribution function which is similar to the standard-normal distribution but easier to work with in most applications (the probabilities are easier to calculate). The logistic distribution constrains the estimated probabilities to lie between 0 and 1.

$$\mathbf{w} \leftarrow \mathbf{w} + \alpha \nabla \log L(\mathbf{w}) = \mathbf{w} + \alpha \sum_{i=1}^{m} (y_i - h_{\mathbf{w}}(\mathbf{x_i}))\mathbf{x_i}$$

where $\alpha \in (0, 1)$ is a step-size or learning rate parameter



SUPPORT VECTOR MACHINE (SVM)

A support vector machine constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks like outliers detection[3]. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier

$$\operatorname{sgn}\left(\sum_{i=1}^{m} \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{x}) + w_0\right)$$

PERCEPTRON

Perceptron is used to separate linearly seperable data.It is a classification algorithm that makes its predictions based on a linear predictor function combining a set of weights with the feature vector. While making the prediction it may not adapt well to the change in data and make a drastic change even when the error is small or vice versa. Here the learning rate is fixed.

$$w_{t+1} \leftarrow w_t + y_t x_t$$

EXPERIMENTAL SETUP

DATASET:

Dataset is constructed by combining benign and malicious URLS taken from web Yahoo directory and Phish tank, Spam websites respectively. Total of 2249 URLS are being used as an input dataset which is 3:2 ratio of benign and malicious URLs respectively. So model will be implemented in such a way that it should be able to isolate Malicious URLS and Benign URLS by extracting Lexical Features and Host-Based Features.

main.py | dataset.csv

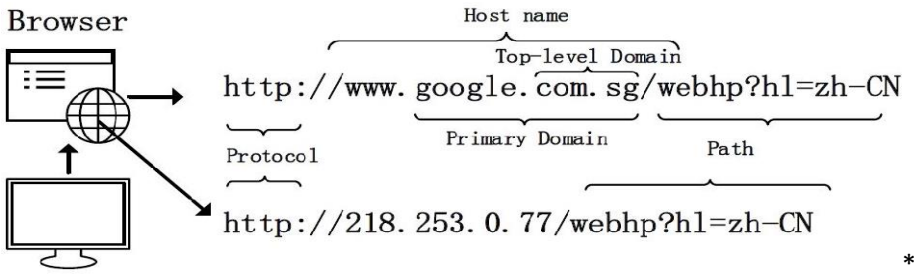
```
1   url,label
2   http://br-ofertasimperdiveis.epizy.com/produto.php?linkcompleto=iphone-6-plus-
3   https://semana-da-oferta.com/produtos.php?id=5abad0c01d149,1
4   https://scrid-apps-creacust-sslhide90766752024.cread-squi.com/hider_reo/,1
5   http://my-softbank-security.com/wap_login.htm,1
6   http://www.my-softbank-security.com/wap_login.htm,1
7   http://diadesaldaolu.infomando.com,1
8   https://sites.google.com/site/helpsettingsrecoveryfbus2018/,1
9   http://protvinowifi.ru/,1
10  http://socset222.96.lt/,1
11  http://my-citibank-security.com/wap_login.htm,1
```

Figure 2 input_dataset.csv

FEATURE EXTRACTION

Lexical Features:

These features exhibits the property that defines how a malicious url tend to look different from benign url. For example, www.google.com contains “com” as a Top-level domain(TLD) which does not look unusual. However, appearance of TLD in www.google.com.sg i.e., “com.sg” could indicate an attempt by the criminal to spoof the domain name of a website. According to the analysis of Malicious URLs, criminals usually try to phish url by adding new words in Top-level Domain. For implementing such features there are some bag of words are used for representation of token in url where‘/’ , ’?’ , ’.’ , ’=’ , ’-’, ’_’ are delimiters. In the project, tokens are being isolated that appears in Top level domain, path, and host-name of malicious URLS and are converted to feature vectors.

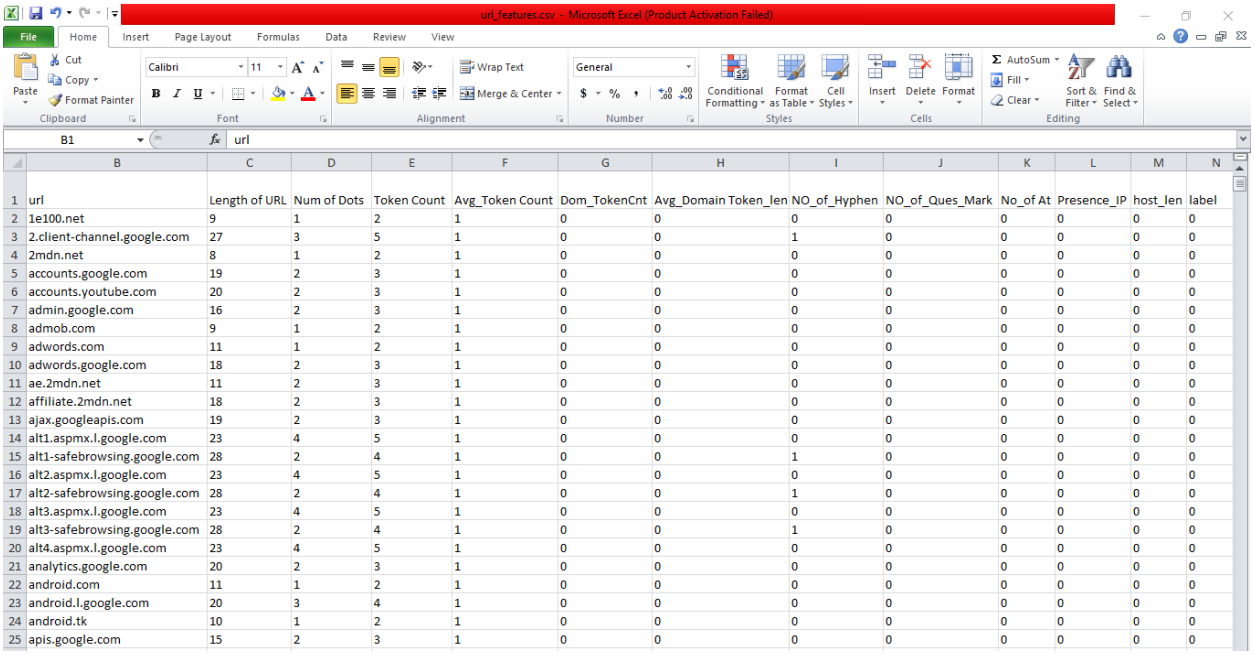


*

Host- based Features:

These features describes the properties of website host, i.e. approximating 'where' these suspicious sites are hosted, 'who' own them and 'how' they are managed. They useful when data is recorded from live feeding of url. Properties like Autonomous System Number (ASN), which has to be unique for every url, Whois information, suspected method usage, Link count of webpage etc.

The project includes total 12 feature vectors: 'url_features.csv'



url_features.csv - Microsoft Excel (Product Activation Failed)

	B	C	D	E	F	G	H	I	J	K	L	M	N
1	url	Length of URL	Num of Dots	Token Count	Avg_Token Count	Dom_TokenCnt	Avg_Domain Token_len	NO_of_Hyphen	NO_of_Ques_Mark	No_of At	Presence_IP	host_len	label
2	1e100.net	9	1	2	1	0	0	0	0	0	0	0	0
3	2.client-channel.google.com	27	3	5	1	0	0	1	0	0	0	0	0
4	2mdn.net	8	1	2	1	0	0	0	0	0	0	0	0
5	accounts.google.com	19	2	3	1	0	0	0	0	0	0	0	0
6	accounts.youtube.com	20	2	3	1	0	0	0	0	0	0	0	0
7	admin.google.com	16	2	3	1	0	0	0	0	0	0	0	0
8	admob.com	9	1	2	1	0	0	0	0	0	0	0	0
9	adwords.com	11	1	2	1	0	0	0	0	0	0	0	0
10	adwords.google.com	18	2	3	1	0	0	0	0	0	0	0	0
11	ae.2mdn.net	11	2	3	1	0	0	0	0	0	0	0	0
12	affiliate.2mdn.net	18	2	3	1	0	0	0	0	0	0	0	0
13	ajax.googleapis.com	19	2	3	1	0	0	0	0	0	0	0	0
14	alt1.aspmx.l.google.com	23	4	5	1	0	0	0	0	0	0	0	0
15	alt1-safebrowsing.google.com	28	2	4	1	0	0	1	0	0	0	0	0
16	alt2.aspmx.l.google.com	23	4	5	1	0	0	0	0	0	0	0	0
17	alt2-safebrowsing.google.com	28	2	4	1	0	0	1	0	0	0	0	0
18	alt3.aspmx.l.google.com	23	4	5	1	0	0	0	0	0	0	0	0
19	alt3-safebrowsing.google.com	28	2	4	1	0	0	1	0	0	0	0	0
20	alt4.aspmx.l.google.com	23	4	5	1	0	0	0	0	0	0	0	0
21	analytics.google.com	20	2	3	1	0	0	0	0	0	0	0	0
22	android.com	11	1	2	1	0	0	0	0	0	0	0	0
23	android.l.google.com	20	3	4	1	0	0	0	0	0	0	0	0
24	android.tk	10	1	2	1	0	0	0	0	0	0	0	0
25	apis.google.com	15	2	3	1	0	0	0	0	0	0	0	0

Figure 3 Features Extracted from URL

RESULTS

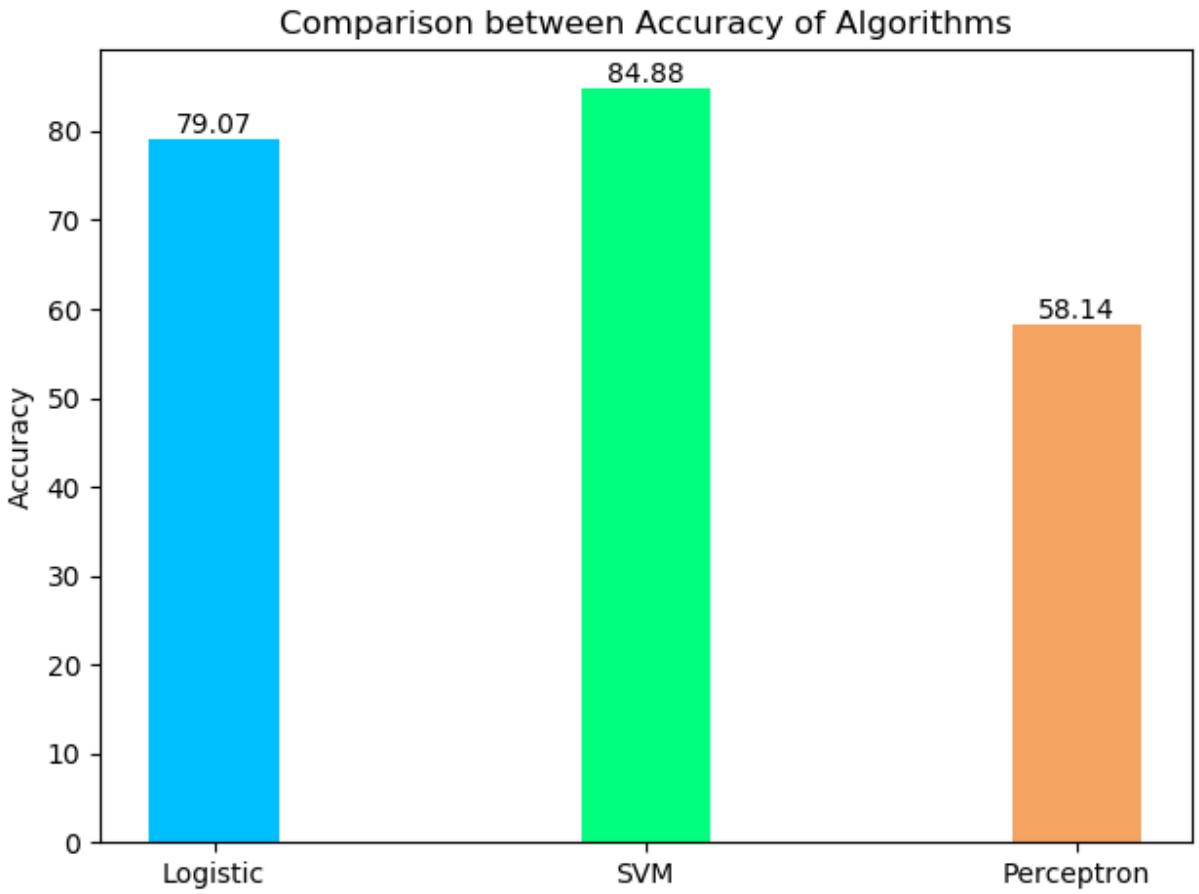


Figure 4 Accuracy obtaned for 10% test data

- Logistic regression performed better than SVM
  with accuracy i.e., 96% with test size = 25%.
- More features = Better accuracy
- From cross validation, it is analyzed that accuracy depends on test size of the dataset i.e., accuracy varies with the increase or decrease in the test size.

Table 1 Comparison of Accuracy and Test size

Test Size (%)	Accuracy(%)		
	Logistic Regression	SVM	Perceptron
10	79	85	58
20	97	81	61
25	96	74	65

CONCLUSION AND FUTURE WORK

Malicious Web sites are a prominent and undesirable Internet scourge. To protect end users from visiting those sites, the identification of suspicious URLs using lexical and host-based features is an important part of a suite of defenses. It is analyzed from the output of the algorithms that,
Future work may include collection of more features to gain a higher accuracy coupled with applying various other classification algorithms.

REFERENCES

- Identifying Suspicious URLs: An Application of Large-Scale Online Learning, Justin Ma, Lawrence K. Saul, Stefan Savage, Geoffrey M. Voelker
- Malicious URL Detection using Machine Learning: A Survey, Doyen Sahoo, Chenghao Liu, and Steven C.H. Hoi
- https://docs.python.org/3/tutorial/datastructures.html
- https://en.wikipedia.org/wiki/Support_vector_machine
- https://stackoverflow.com/questions/44890713/selection-with-loc-in-python
- http://python-graph-gallery.com/barplot/
- Sources of malicious and benign URLs
  - *http://apac.trendmicro.com/apac/security-intelligence/current-threat-activity/malicious-top-ten/*
  - *https://www.phishtank.com/phish_archive.php*
  - *http://www.malwaredomains.com*
  - *https://db.aa419.org/fakebankslist.php?start=21*