Malicious URL Detection Using Several Classification Machine Learning Algorithms
COMP 562 Intro to Machine Learning - Final Project

Yiyang Chen, Qiyang Huang, Ruochen Li, Yuyang Sun

December 15th 2019

Abstract

Project link: https://github.com/liruochen1998/MaliciousURLDetection

We use 6 different machine learning algorithms, logistic regression, SVM, Lasso regression, Naive Bayes, Random Forest, and Decision Tree in identifying whether a given URL is malicious. From the six strategies, we learned first four in class and the last two by self-learning and some research. We have 1 million URL data with label indicating whether it's malicious, and we use 11 features for each URL data.

1 Introduction

Nowadays, the internet is playing a much more important role in our life and we use it everywhere. However, risks follow. A Uniform Resource Locator (URL), colloquially termed a web address, is a reference to a web resource that specifies its location on a computer network and a mechanism for retrieving it. Simply put, receiving a URL can be similar to a stranger inviting you to their house. Their invitation might promise food and drink, and you could go over for a visit, but you have no idea what will really happen until you walk inside. Thus, if we click on the URLs given without checking it, we may be led to a malicious website. These URLs are called malicious URLs, which lead to websites that may include spam, malware, and phishing.

In this project, we use different machine learning methods to identify the malicious URLs. In order to find out the best model to deal with the problem, we choose six classification methods. Four of them are introduced during the class and we use two extra popular methods that are believed to have the ability to provide good performance. We use the data collected from seven different online resources that contain different kinds of malicious URLs(spam, malware, phishing) to establish and test the models. At last, we compared their performance and have a discussion.

2 Data Selection

2.1 Data Format

The data are collected from seven different databases including websites that are involved in phishing, malware, and spam. And we collect list of benign URLs and add into the dataset for training data. There are 1 million URLs in the dataset that we are using in total, among which 15% are malicious while the rest are benign. There are two columns for each entry of the original dataset, which is the URL itself and a label, indicating malicious(label being 1) and benign(label being 0). Thus, because the output is a binary indicator (either a 0 or 1), this is a classification problem. And

from the original, we are going to select some features based on the characteristic of URLs.

2.2 Feature Selection

The syntax for every valid URL is

```
URI = scheme:[//authority]path[?query]
[#fragment] Where authority can also be
divided into three subcomponents:
authority = [userinfo@]host[:port].
```

Thus, we can use the syntax to generate some features for the dataset. From the syntax, we believe there are some significant differences in malicious URLs and benign URLS, such as length, the appearance of some specific symbols, or the structures, etc. In this research project, specifically, we focus on 10 distinct characteristics for each URL and use the 10 characteristics as 10 features for the dataset.

Firstly, there are features about length:

Length of url: We believe the length of the malicious will be substantially longer than that of the benign urls, which is the reason why we use the length of the whole url as the feature of the dataset.

Length of domain name: The domain name is part of the url, which means there might be a relationship between the length of the domain name with the length of the url, but only the length of the url might not be sufficient for identifying a malicious url. Thus, we also want to take length of domain name into account.

Secondly, a very important component of the url is the domain name, and since there is a hierarchical structure for domains, the top level domain (TLD) will be very important for identifying a URL. We collected 10 most abused TLDs. For example, the most abused TLD from 2019 is .buzz. There is 32,000 websites using .buzz TLD in total, and about 18,000 of them are malicious websites. Thus, we use whether the website contains the 10 most abused TLDs as a feature for classified the data.

Thirdly, malicious urls commonly have more symbols, such as delimiters

```
(;, _, ?, =, and &)
```

, and by the structure of the url, some symbols have special meanings. For example, double-slash(//) might be used to redirect you to a new ip address or url, @ might be used to username:password pair, and ? might be used to indicate the start of a query. Thus, we also have the following feature selection:

Number of slashes: this also has another meaning, which is the number of subdirectories.
Number of subdomains: we use the number of the extension to count the number of subdomains
Number of queries
Number of double slashes
Number of @ symbol
Number of - symbol
Number of . in subdomain

Lastly, there is some url, which is formed directly by an IP, which we will also take into consideration.

Thus, there are 11 features in the dataset in total.

3 Methods

This is a classification question, therefore we choose six typical classification methods that are used nowadays, which are logistic regression, Gaussian naive Bayes, SVM, Lasso regression, decision tree, and random forest.

3.1 Logistic Regression

Logistic Regression is a Machine Learning algorithm which is used for classification problems, it is a predictive analysis algorithm and based on the concept of probability, compared to linear regression, Logistic Regression uses a more complex cost function, this cost function can be defined as the 'Sigmoid function'. Also, we add ridge penalty for the logistic regression for dealing with potential col-linearity issue.

3.2 Naive Bayes

Naive Bayes is a simple, yet effective and commonly-used, machine learning classifier. It is a probabilistic classifier that makes classifications using the Maximum A Posterior decision rule in a Bayesian setting with the assumption that every event are independent. It can also be represented using a very simple Bayesian network. Naive Bayes classifiers have been especially popular for text classification, and

are a traditional solution for problems such as spam detection. There are different kind of Naive Bayes that can be applied, here we use Naive Bayes with Gaussian function specifically.

3.3 Support Vector Machine

Support vector machine (SVM) is an algorithm that aim to find a hyperplane in N-dimensional space (N — the number of features) that distinctly classifies the data points, a hyperplane that has the maximum margin, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

3.4 Lasso Regression

Lasso regression is based on logistic regression, but instead of using ridge penalty, which uses the L2 term that is equal to the square of the magnitude of the coefficients, lasso regression uses the L1 penalty term and stands for Least Absolute Shrinkage and Selection Operator. In previous research of the similar topic, people seldom use Lasso regression. We intend to include it here to see whether this can be a good algorithm for solving this problem.

3.5 Decision Tree

Decision tree is a tree-based algorithm which uses a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements. A decision tree is a flowchart-like structure in which each internal node represents a "test" on an attribute. The paths from the root to the leaf represent classification rules.

3.6 Random Forest

Random forest is based on decision tree. It consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction. It correct for decision trees' habit of overfitting to their training set.

3.7 Training and Testing

We are going to use the six strategies and algorithms as the methods for this classification problem. For the whole data set, we use cross validation for training and testing. This means that we are picking a portion of the dataset as training set and the rest as testing set, and pick the other portion as training and the rest as testing, etc.

4 Results

This is the score for the 6 different algorithms for the classification problem:

DecisionTree	0.99877
RandomForest	0.99886
GaussianNaiveBayes	0.99810
LogisticRegression	0.99858
SVM	0.99867
LassoRegression	0.71825

Among all the 6 strategies here, each method has a very good prediction of the sample except Lasso regression. The detailed score is shown above in the chart. The score here is unique for each of the machine learning algorithm, which means different methods have a different way of scoring, but they do have meaningful interpretation and can be compared together.

From the result we can see that among SVM, Naive Bayes, Logistic Regression, and Lasso, SVM has the best performance, while Lasso has the worst performance, only a score of over 0.7. Decision Tree and Random Forest, which we haven't learned in class, have slightly better performance than SVM. Excluding Lasso, none of them are significant better than the others, but they do vary in speeds when dealing with large dataset.

5 Discussion

All the methods are quite successful(success rate exceeds 99.8%)except lasso regression.

The reason behind lasso regression's failure is probably because lasso regression is designed for sparse data sets(eg. spam email classifier). For this dataset, we have 11 features in total. Even though for each entry of the dataset (each URL), one usually has 2 to 3 meaningful entries, which is non-zero features, this still can't be considered sparse. This data training set does not meet the requirement of a desired sparse dataset, therefore lasso regression shall not be used when training this kind of classifier model.

The rest five strategies are very good methods for classification problem like this, identifying malicious URL.

The future research area about this topic can be the following. The most important one is what is the best way to select features for URLs. Here, we use these 11 features because of the syntax and structure of URLs, but they are definitely not all the meaningful features that they can generate. For example, one feature that might be meaningful is the time to expire of each URL using WHOIS information. People who own benign websites tend to keep the websites up to date and probably don't want it expire.

From this topic, we can come up with some other research area for future studies, which might include how the HTML page of a websites be identified as malicious.

We believe this research, identifying malicious URL using these six machine learning strategies have a very meaningful not only in real life but also in academia.

6 References

Justin Ma , Lawrence K. Saul , Stefan Savage , Geoffrey M. Voelker, Learning to detect malicious URLs, ACM Transactions on Intelligent Systems and Technology (TIST), v.2 n.3, p.1-24, April 2011
Justin Ma , Lawrence K. Saul , Stefan Savage , Geoffrey M. Voelker, Beyond blacklists: learning to detect malicious web sites from suspicious URLs, Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, June 28-July 01, 2009, Paris, France

https://www.vircom.com/blog/what-is-a-malicious-url/
https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148
https://towardsdatascience.com/introduction-to-naive-bayes-classification-4cffabb1ae54
https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47
https://medium.com/greyatom/decision-trees-a-simple-way-to-visualize-a-decision-dc506a403aeb
https://towardsdatascience.com/understanding-random-forest-58381e0602d2
https://docs.google.com/document/d/$1_E w5cilaCvxIqB3P50Qsxfz_r Cdt - JBaiBwEyktXCgY/edit$
$https://hackernoon.com/an-introduction-to-ridge-lasso-and-elastic-net-regression-cca60b4b934f$
$https://www.spamhaus.org/statistics/tlds/$