Malicious URL Detection based on Machine Learning

Nikita Bhamu (2018CS50413), Manoj Kumar (2018CS50411)

April 2, 2022

1 Abstract

With the increase in internet usage, the danger to its security has also increased. Cybersecurity has emerged as a big challenge in recent times. The attackers use different ways to attack end-to-end technology to get more information about the user, which can lead to the user's exploitation afterward. One such way to deceive the users is using malicious URLs.These malicious URLs perform various deceitful exercises, such as unauthorized private and secret information access. So, it has become essential to detect and make users aware of a malicious URL before they click it based on the previous historical record of the URLs present. Several research papers have demonstrated various approaches for detecting malicious URLs using machine learning and deep learning techniques.
In this project, we tend to look at the various features that can help us classify the URL as malicious or benign, and then we tend to look at the importance of those different features.
Then we planned to classify the input sets of URLs using five different supervised machine learning techniques, which includes Random forest, Support Vector Machine, Gradient Boost classifier, Logistic Regression, Naive Bayes, and then compare the performance of all these algorithms in the malicious URL detection.

2 Introduction

Among the 10 most frequent attack strategies, attacks utilising the spreading malicious URL strategy are ranked top. The three primary URL spreading techniques, malicious URLs, network URLs, and phishing URLs, are seeing an increase in both the number of attacks and the level of threat.
Based on the increasing number of malicious URL distributions throughout the years, it is evident that techniques or procedures to detect and prevent these malicious URLs are needed to be studied and used.
A URL is made of two basic components: Protocol Identifier (Protocol type)

and Resource Name (IP address or domain name).

There are two primary trends in malicious URL detection at the moment: malicious URL detection based on signs or sets of rules, and malicious URL detection based on behaviour analysis techniques. The approach of detecting malicious URLs based on a set of markers or rules may detect malicious URLs rapidly and reliably. This approach, on the other hand, is incapable of detecting new malicious URLs that do not match the specified signs or rules. Machine learning or deep learning algorithms are used to classify URLs based on their behaviours in the process of detecting malicious URLs based on behaviour analysis techniques.

Machine learning algorithms are used in our study to classify URLs based on their features and behaviours. The features are extracted from the static and dynamic behaviour of URLs. The research's key contribution is the newly proposed features. The whole malicious URL detection system involves machine learning techniques.

The attackers change one or more components of URLs and redirect the URLs to malicious resources. They can execute code on users' computers to malware download or redirect them to unwanted and other phishing sites. A sudden increase in the number of such malicious URLs has been observed in the last few years. Hence, there is a need to study some techniques to detect and prevent these malicious URLs.

Here we use the following machine learning models on the lexical features of URLs to predict whether the test URL is malicious or not:

- Logistic Regression with TfIdf and tokenization.
- Neural Network model on lexical features.
- Logistic Regression on lexical features.
- Random Forest on Lexical Features.
- Decision Tree on Lexical Features.

3 Background and Related Work

Many antivirus groups have worked in this field to detect and prevent the malicious URLs.

1. **Signature based Malicious URL Detection / Blacklist method:** The most common method to detect malicious URLs deployed by many antivirus groups is the blacklist method. It detects using the signature

sets of malicious URLs. A database query is executed to find that URL whenever a new URL is accessed. If that URL exists in the database, that means that the URL is blacklisted. Hence, it is considered a malicious URL; otherwise, it is considered safe. This is the commonly used method by most companies, but the main disadvantage of this approach is that it will be complicated to detect the new malicious URLs that do not exist in our database.

2. **Machine Learning Based Malicious URL Detection:** Machine Learning algorithms use a set of URLs as training data and learn a prediction function to classify a URL as malicious or safe based on the statistical properties of the data. It removes the disadvantage of the blacklisting method as it also predicts for new URLs.
   The Machine Learning algorithms can be classified into supervised (when we have labels for training data), unsupervised (we do not have labels), and semi-supervised (we have labels for a limited fraction of training data) algorithms. Plenty of classification algorithms can be used directly over the training data, such as naive Bayes, Support Vector Machine, Logistic Regression, etc. But the scalability and efficiency of these algorithms depend on the features and size of the training sets.

3. **Malicious URL Detection Tools:** There are some tools available on the internet that detects malicious URLs.

   - **URL Void:** It is a URL checking program that uses multiple engines and blacklists of domains. The main advantage of this tool is its compatibility as it supports many testing services and browsers. The main disadvantage of this tool is that it depends more on the given set of signatures.
   - **UnMask Parasites:** This tool downloads the data of provided links, parses the HTML codes, javaScript, external links and iframes and analyze if something strange is happening on the site.
   - **Dr. Web Anti-Virus Link Checker:** It is an add-on for browsers like Chrome, Firefox, Opera and Microsoft Edge to automatically find and scan malicious content on a download link on all social networking links such as Facebook, Google+ etc.
   - **Some Other tools**: There are some more URL checking tools such as UnShorten.it, VirusTotal, Norton Safe web, McAfee SiteAdvisor, Google Safe Browsing etc.

4 Problem Statement

The problem here is a classification problem of the URLs given as input. To be precise, it can be seen as a binary classification problem that classifies the URL either as malicious(-1) or as benign(+1).

And this classification is to be accomplished based on the already labeled training data of a sufficiently large number of URLs in which all the URLs are labeled with +1 if those are benign and -1 if those are malicious.

This URL detection can be achieved in two folds which are the following:

1. Extract the various features from the URL that play a role in the classification, trying to develop some new features that can be of use—finally representing all the URLs in those feature vectors.
2. Employ different machine learning algorithms to classify the URLs based on the feature vectors we got in the first part.

We have an analysis factor attached to both the folds where :-

1. In the first fold, we will analyze the importance of the features used.
2. In the second fold, we will analyze the relative performance of the machine learning algorithms used, which is their predictive accuracy.

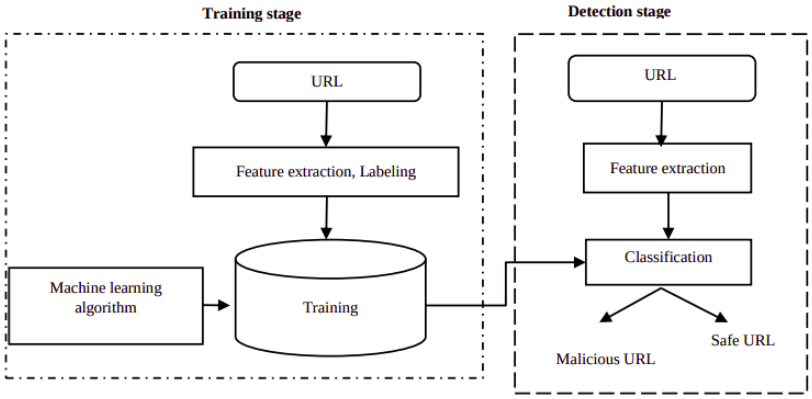
5 Proposed Solution:

Extraction of Features :-

We will consider the following types of features :

- Lexical features :- Length of the URL, the length of each URL component (Domains,subdomains, path), number of special characters, and each character sequence separated by a special character ("/", ". ")
- Content Based :- We can get the HTML code of the page and analyze the number of words, average words per line, distinct words, links to remote scripts and invisible objects. Other features we can get are related to JavaScript as they are used by hackers to encrypt malicious code or execute without permission.
- Measures of popularity :- Link Popularity which is scored based on incoming links from other webpages. Also used are the number of popups and the behavior of plugins.

Here, we want to apply machine learning algorithms on millions of training data, and extracting the content based features of all training URLs is neither memory feasible not speed efficient. Hence, to train our ML models, we will use the **lexical features** only.



We are using the following lexical features to train the ML models:

- number of character '.' in URL
- number of subdomain levels
- The depth of URL
- The length of URL
- Number of the dash character ''
- Number of dash character in the hostname
- There exists a character '@' in URL
- There exists a character ~ (tilde) in URL
- Number of the underscore character
- Number of the character '%'
- Number of the character '&'
- Number of the character '#'
- Number of the numeric characters
- Check if the IP address is used in the hostname of the website URL
- Length of hostname
- Length of the link path
- Number of sensitive words (i.e., "secure", "account", "webscr", "login", "ebayisapi", "sign in", "banking", "confirm") in website.

Machine Learning Algorithms :-

We trained different ML models (Neural Network, Logistic Regression, Decision Trees, Random Forests etc) with the above mentioned features.

We also train our logistic regression model with the *tokenization and vectorization* of the URL using **TfIdf**.

So there are total of 5 ML models:

- **Logistic Regression with TfIdf**: It is a well-known machine learning model which computes the conditional probability for a feature vector x of a URL to be classified as a class y = 1, which denotes it to be malicious by the probability function given below.

  $$P(y = 1|\mathbf{x}; \mathbf{w}, b) = \sigma(\mathbf{w} \cdot \mathbf{x} + b) = \frac{1}{1 + e^{-(\mathbf{w} \cdot \mathbf{x} + b)}}$$

  On our dataset, we extract the words in the URL and applied Tf-Idf model to extract features.

- **Neural Network with Lexical Features hidden layers=(20, 10):** Neural networks are comprised of a node layers. The model contains an input layer, one or more hidden layers, and an output layer. Each node, or artificial neuron, connects to another and has an associated weight. The data passed to the next node is based on the weight of the edge between both nodes of consecutive layers.
  Here we take 2 hidden layers with hidden layer with 20 nodes and second hidden layer with 10 nodes.

- **Logistic Regression on Lexical Features:** This time, we use the lexical features of URL as the input of the model.

- **Random Forests on Lexical Features:** Random Forest (RF) is the ensemble classifier, which collects the results of many decision trees by majority vote.
  For this classification tasks, the output of the random forest is the class selected by most trees, i.e., if most of the decision trees classify the URL to be malicious based on the feature vector of it then the URL is considered to be malicious.

- **Decision Tree with Lexical Features:** he decision tree is like a tree with nodes. The branches depend on a number of factors. It splits data into branches like these till it achieves a threshold value.

6 Evaluation:

The dataset for this project is taken from Kaggle (`https://www.kaggle.com/datasets/antonyj453/urldataset`). The dataset has two attributes: URL and label. There are 4,20,000 rows.

We are randomly taking 80% of the data as our training set and testing on the remaining 20% data.

We evaluate the results based on the following parameters:

- Accuracy of the model
- Confusion Matrix
- Classification report

The evaluation report on various ML models is as follows:

- **Logistic Regression with TfIdf**:

```
Accuracy of our model is : 0.9645630432973018
[[12323  2641]
 [  339 68790]]
              precision    recall  f1-score   support

         bad       0.97      0.82      0.89     14964
        good       0.96      1.00      0.98     69129

    accuracy                           0.96     84093
   macro avg       0.97      0.91      0.94     84093
weighted avg       0.96      0.96      0.96     84093
```

- **Neural Network with Lexical Features hidden layers=(20, 10):**

```
Accuracy of our model is : 0.9000511338636985
[[ 9438  5699]
 [ 2706 66250]]
              precision    recall  f1-score   support

       False       0.78      0.62      0.69     15137
        True       0.92      0.96      0.94     68956

    accuracy                           0.90     84093
   macro avg       0.85      0.79      0.82     84093
weighted avg       0.89      0.90      0.90     84093
```

- **Logistic Regression on Lexical Features:**

```
Accuracy of our model is : 0.8538403909956833
[[ 3556 11637]
 [  654 68246]]
              precision    recall  f1-score   support

       False       0.84      0.23      0.37     15193
        True       0.85      0.99      0.92     68900

    accuracy                           0.85     84093
   macro avg       0.85      0.61      0.64     84093
weighted avg       0.85      0.85      0.82     84093
```

- **Random Forests on Lexical Features:**

```
Accuracy of our model is : 0.9224905759100044
[[11131  4020]
 [ 2498 66444]]
              precision    recall  f1-score   support

       False       0.82      0.73      0.77     15151
        True       0.94      0.96      0.95     68942

    accuracy                           0.92     84093
   macro avg       0.88      0.85      0.86     84093
weighted avg       0.92      0.92      0.92     84093
```

- Decision Tree with Lexical Features.

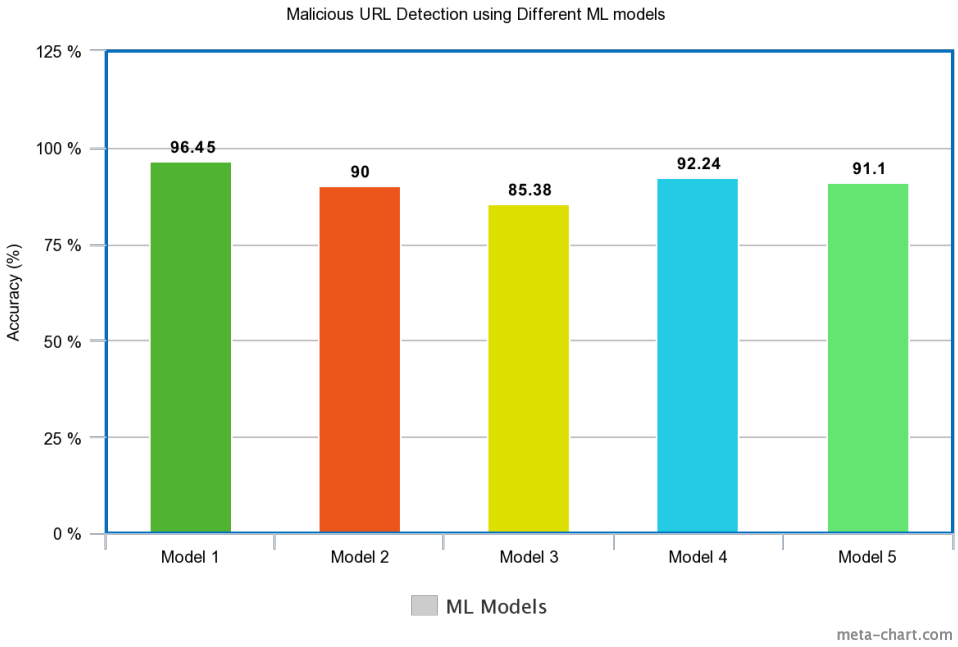
```
Accuracy of our model is : 0.9110151855683588
[[10671  4339]
 [ 3144 65939]]
              precision    recall  f1-score   support

       False       0.77      0.71      0.74     15010
        True       0.94      0.95      0.95     69083

    accuracy                           0.91     84093
   macro avg       0.86      0.83      0.84     84093
weighted avg       0.91      0.91      0.91     84093
```

Complete python code of evaluation is uploaded here: `https://github.com/manoj2601/Malicious-URL-Detection-using-Machine-Learning`

7 Conclusion and Future Work

We got the following accuracy percentage for different ML models on the same dataset:

ML Model No.	Description	Accuracy (%)
Model 1	Logistic Regression with TfIdf	96.45
Model 2	Neural Network with Lexical Features (hidden-layer size=(20,10))	90%
Model 3	Logistic Regression on Lexical Features	85.38%
Model 4	Random Forests on Lexical Features	92.24%
Model 5	Decision Tree with Lexical Features	91.10%



Here we observe that logistic regression model with tokenization and Tf-Idf is most accurate model for the given dataset.

Although, all of the models are with a good accuracy. Detection of Malicious URLs plays a critical role in many cybersecurity applications, and machine learning approaches are a promising direction. There is a wide variety of data that can be obtained from a URL. Transforming this data to a machine learning compatible feature vector can be very resource intensive and it can improve the predictive models.

Machine Learning has made the prediction for the malicious URLs more easy and now it does not limited to the limited signature set. We are now capable to make predictions on the newer malicious URLs as well and these predictions are more often accurate.

The malicious URL detection tool can be very useful for our daily browsing as many data leak incidents are happening now a days. In future, the broswers might also introduce the built-in malicious URL detection feature to make browsing more secure.

8 References

[1] D. Sahoo, C. Liu, S.C.H. Hoi, "Malicious URL Detection using Machine Learning: A Survey". CoRR, abs/1701.07179, 2017.

[2] M. Khonji, Y. Iraqi, and A. Jones, "Phishing detection: a literature survey," IEEE Communications Surveys Tutorials, vol. 15, no. 4, pp. 2091–2121, 2013.

[3] M. Cova, C. Kruegel, and G. Vigna, "Detection and analysis of drive-by-download attacks and malicious javascript code," in Proceedings of the 19th international conference on World wide web. ACM, 2010, pp. 281–290.

[4] R. Heartfield and G. Loukas, "A taxonomy of attacks and a survey of defence mechanisms for semantic social engineering attacks," ACM Computing Surveys (CSUR), vol. 48, no. 3, p. 37, 2015.

[5] S. Sheng, B. Wardman, G. Warner, L. F. Cranor, J. Hong, and C. Zhang, "An empirical analysis of phishing blacklists," in Proceedings of Sixth Conference on Email and Anti-Spam (CEAS), 2009.

[6] C. Seifert, I. Welch, and P. Komisarczuk, "Identification of malicious web pages with static heuristics," in Telecommunication Networks and Applications Conference, 2008. ATNAC 2008. Australasian. IEEE, 2008, pp. 91–96.

[7] S. Sinha, M. Bailey, and F. Jahanian, "Shades of grey: On the effectiveness of reputation-based "blacklists"," in Malicious and Unwanted Software, 2008. MALWARE 2008. 3rd International Conference on. IEEE, 2008, pp. 57–64.

[8] Pelin Zhao, Steven C.H.Hoi, "Cost-Sensitive Online Active Learning with Application to Malicious URL Detection", Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 11-14, 2013, Chicago. 919-927. Research Collection School Of Information Systems

[9] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Identifying suspicious urls: an application of large-scale online learning," in Proceedings of the

26th Annual International Conference on Machine Learning. ACM, 2009, pp. 681–688.

[10] Rakesh Verma, Avisha Das, "What's in a URL: Fast Feature Extraction and Malicious URL Detection" proceeding IWSPA '17 Proceedings of the 3rd ACM on International Workshop on Security and Privacy Analytics Pages 55-63.

[11] Christophe Chong, Daniel Liu, Wonhong Lee, "Malicious URL Detection" Published at Stanford University, with Neustar, 2012.

[12] R.k. Nepali and Y. Wang "You Look suspicious!!" Leveraging the visible attributes to classify the malicious short URLs on Twitter. in 49th Hawaii International Conference on System Sciences(HICSS) IEEE, 2016, pp. 2648-2655.

[13] Doyen Sahoo, Chenghao Liu and Steven C.H.Hoi "Malicious URL Detection using Machine Learning A Survey", 2016, an article in the arxiv.

[14] B. Eshete, A. Villafiorita, and K. Weldemariam, "Binspect: Holistic analysis and detection of malicious web pages," in Security and Privacy in Communication Networks. Springer, 2013, pp. 149–166.

[15] Mohammed Al-Janabi, Ed de Quincey, Peter Andras, "Using Supervised Machine Learning Algorithms to Detect suspicious URLs in online social networks", Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017

[16] S. Purkait, "Phishing counter measures and their effectiveness– literature review," Information Management Computer Security, vol. 20, no. 5, pp. 382–420, 2012.

[17] Y. Tao, "Suspicious url and device detection by log mining," Ph.D. dissertation, Applied Sciences: School of Computing Science, 2014.

[18] G. Canfora, E. Medvet, F. Mercaldo, and C. A. Visaggio, "Detection of malicious web pages using system calls sequences," in Availability, Reliability, and Security in Information Systems. Springer, 2014, pp. 226–238.

[19] Leo Breiman.: Random Forests. Machine Learning 45 (1), pp. 5- 32, (2001).

[20] Thomas G. Dietterich. Ensemble Methods in Machine Learning. International Workshop on Multiple Classifier Systems, pp 1-15, Cagliari, Italy, 2000.