Malicious URL Detection using Machine Learning with Feature Extraction

Harsh Dungrani[1], Yagnesh Gotad[2]
Electronics and Telecommunication Engineering Department,
Sardar Patel Institute of Technology,
Mumbai, India

I. Introduction

Malicious websites are common and serious threat to cybersecurity. Malicious URLs host unsolicited content like Spam, Phishing and drive-by exploits. Lure unsuspecting users to become victims of scams (monetary loss, theft of private information, and malware installation), and cause losses of billions of dollars every year. Imperative to detect and act on such threats in a timely manner. Uniform Resource Locator (URL) is used to refer to resources on the Internet. Attackers often try to change one or more components of the URL's structure to deceive users for spreading their malicious URL. Malicious URLs are known as links that adversely affect users. These URLs will redirect users to resources or pages on which attackers can execute codes on users' computers, redirect users to unwanted sites, malicious websites, or other phishing sites, or malware download. Malicious URLs can also be hidden in download links that are deemed safe and can spread quickly through file and message sharing in shared networks.

Traditionally, detection is done mostly through the usage of blacklists. However, blacklists cannot be exhaustive, and lack the ability to detect newly generated malicious URLs. To improve the generality of malicious URL detectors, machine learning techniques have been explored with increasing attention in recent years.

Regarding the problem of detecting malicious URLs, there are two main trends at present as malicious URL detection based on signs or sets of rules, and malicious URL detection based on behavior analysis techniques. The method of detecting malicious URLs based on behavior analysis techniques adopt machine learning algorithms to classify URLs based on their behaviors.

In our project, we have carried out a comparative analysis of various different machine learning models like Logistic Regression, Random Forest, Decision Tree, KNN, Linear Discriminant Analysis, Artificial Neural Networks and Support Vector Machines. First feature extraction was done to get lexical features such as the length features, count feature and binary features.

II. Methodology

A. Dataset

The dataset initially consisted of 3 columns:

1. url(str): name of the url
2. label(str): benign or malicious
3. result(int): 0 or 1

This dataset was then extended into the one having 22 columns using extensive feature extraction. The dataset was then converted with various lexical attributes of the urls. They are as follows:

1. url(str): name of the url
2. label(str): label of the url; malicious or benign
3. result(int): 1 and 0
4. url_length(int): length of the url
5. hostname_length(str): hostname of the server
6. path_length(int): length of the path
7. fd_length(int): length of the first directory
8. tld_length(int): length of the top level domain
9. count-(int): count of '-' in the url
10. count@(int): count of '@' in the url
11. count?(int): count of '?' in the url
12. count%(int): count of '%' in the url
13. count.(int): count of '.' in the url
14. count=(int): count of '=' in the url
15. count-http(int): count of 'http' in the url
16. count-https(int): count of 'https' in the url
17. count-www(int): count of 'www' in the url
18. count-digits(int): count of digits in the url
19. count-letters(int): count of letters in the url
20. count_dir(int): number of directories in the url
21. use_of_ip(binary): if ip used or not
22. short_url(binary): if url shortened or not

Out of which only 18 features were used in model prediction, excluding the name of url, label, url_length and short_url features. The total size of the dataset after feature extraction was 470176 rows into 18 columns. The target variable was the result attribute where 0 is benign and 1 being malicious.

B. URL Parsing

Various lexical features of the url were obtained using the urllib.parse and tld libraries in python.

1. *tld*: Top level domain is a python library used to extract the top level domain (TLD) from the URL given. List of TLD names is taken from Public Suffix. Optionally raises exceptions on non-existing TLDs or silently fails (if fail_silently argument is set to True).

2. *urllib.parse*: This module defines a standard interface to break Uniform Resource Locator (URL) strings up in components (addressing scheme, network location, path etc.), to combine the components back into a URL string, and to convert a "relative URL" to an absolute URL given a "base URL." The module has been designed to match the internet RFC on Relative Uniform Resource Locators. It supports the

following URL schemes: file, ftp, gopher, hdl, http, https, imap, mailto, mms, news, nntp, prospero, rsync, rtsp, rtspu, sftp, shttp, sip, sips, snews, svn, svn+ssh, telnet, wais, ws, wss.

C. Machine Learning Models

1. *Logistic Regression:* Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. The nature of the target or dependent variable is dichotomous, which means there would be only two possible classes. In simple words, the dependent variable is binary in nature having data coded as either 1 (stands for success/yes) or 0 (stands for failure/no).

2. *Gaussian Naive Bayes:* Naive Bayes is a simple but effective classification technique which is based on the Bayes Theorem. It assumes independence among predictors, i.e., the attributes or features should not be correlated to one another or should not, in any way, be related to each other. Even if there is dependency, still all these features or attributes independently contribute to the probability and that is why it is called Naïve.

3. *Support Vector Machine:* Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future.

4. *Decision Tree:* Decision tree is a supervised learning algorithm. This technique is mostly used in classification problems. It performs effortlessly with continuous and categorical attributes. This algorithm divides the population into two or more similar sets based on the most significant predictors. Decision Tree Algorithm, first calculates the entropy of each and every attribute. Then the dataset is split with the help of the variables or predictors with maximum information gain or minimum entropy. These two steps are performed recursively with the remaining attributes.

5. *Random Forest:* Random Forest is also a popularly supervised machine learning algorithm. This technique can be used for both regression and classification tasks but generally performs better in classification tasks. As the name suggests, Random Forest technique considers multiple decision trees before giving an output. So, it is basically an ensemble of decision trees. This technique is based on the belief that more trees would converge to the right decision.

6. *Linear Discriminant Analysis:* Linear Discriminant Analysis is a dimensionality reduction technique which is commonly used for the supervised classification problems. It is used for modeling differences in groups i.e. separating two or more classes. LDA makes some simplifying assumptions about the data: 1. That the data is Gaussian (Normal) that each variable is shaped like a bell curve when plotted. 2. That each attribute has the same variance that values of each variable vary around the mean by the same amount on average

7. *Neural networks:* A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. In this sense, neural networks refer to systems of neurons, either organic or artificial in nature. Neural networks can adapt to changing input; so the network generates the best possible result without needing to redesign the output criteria. In Neural Networks, three different optimizers namely "Adam", "Adamax" and "Nadam" are used and keeping the number of epochs as 10 the performance of the three classifiers is analyzed.

8. *AdaBoost CLassifier:* An AdaBoost classifier is a meta-estimator that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases.

III. Results And Discussion

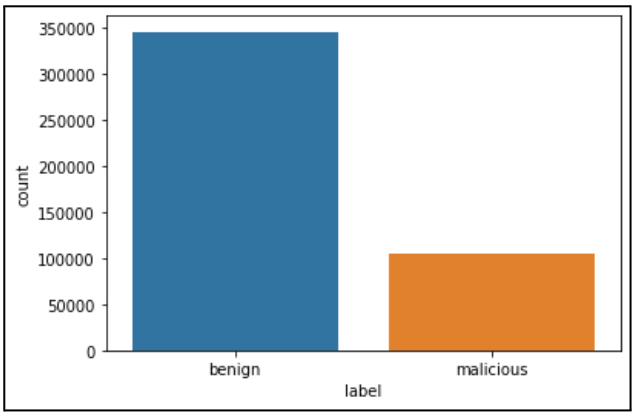

Fig 1: Label count of urls

The bar chart shown in Fig 1, depicts the number of benign and malicious urls. It shows that there is an imbalance in data.

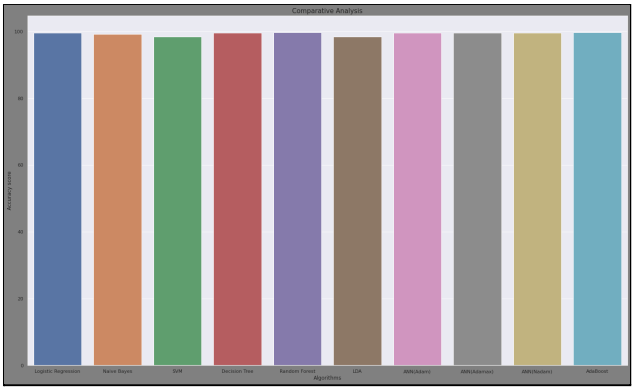

Fig 2: Comparative Analysis of the models

The graph in Fig 2 depicts accuracy metric comparison of the models used. It can be used that all the models used have an accuracy above 95%. This high accuracy can be attributed to the feature extraction process carried on the dataset.

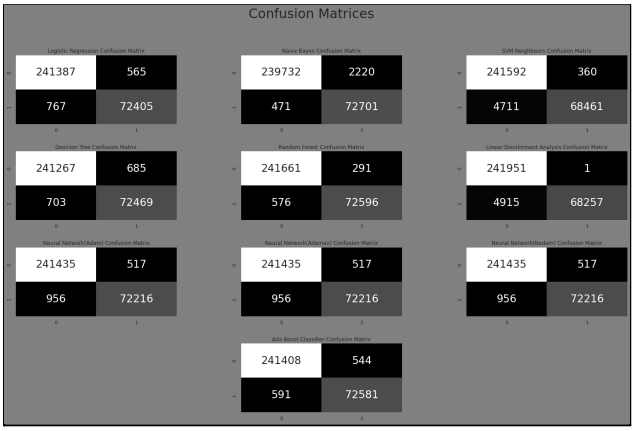

Fig 3: Confusion matrix

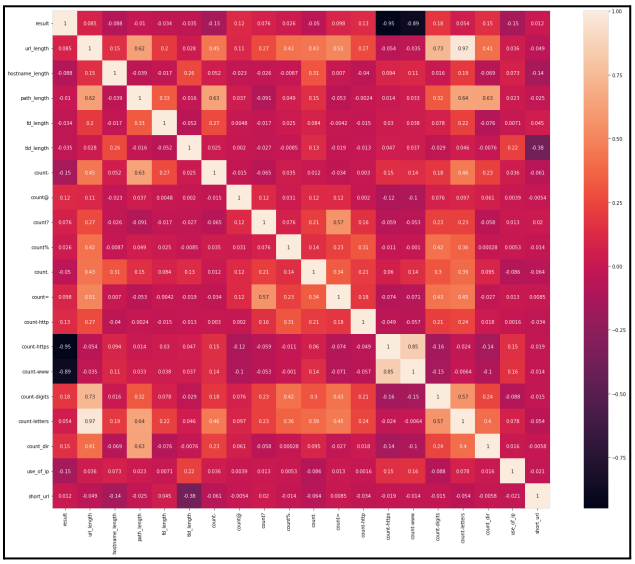

Fig 4: Correlation Matrix

Fig 3 depicts the confusion matrices of all the machine learning models used.
Fig 4 shows the correlation matrix of the features. From the correlation matrix it can be seen that the count-https and count-www features are very highly negatively correlated to the result attribute.

Models	Accuracy Scores
Logistic Regression	99.57
Naive Bayes	99.14
SVM	98.39
Decision Trees	99.55
Random Forest	99.72
LDA	98.43
ANN Adam	99.53
ANN Adamax	99.55
ANN Nadam	99.51
Ada Boost	99.63

Table 1: Accuracy Scores of models

Table 1 shows the model metrics when the entire dataset is trained and predicted using various machine learning classifiers.

The proposed Random Forest Model has the best performance parameters when compared to other machine learning models on the given dataset. They are as follows:

1. Accuracy: 0.9972487020982217
2. Recall Score: 0.9972487020982217
3. Precision Score: 0.9972472418038443
4. F1 Score: 0.9972468275658526
5. ROC AUC Score: 0.9954627092755882.

IV. Conclusion

The work used various feature extraction techniques to use the lexical features of the usps including the length features, count features and the binary features. The dataset initially consisted of only 3 attributes namely name or the url, label and the result. These parameters were not enough for the model to train and predict upon. So an extensive feature extraction was carried which extracted lexical features such as length, count and binary attributes related to the url. After the feature extraction the dataset consisted of 17 input variables: hostname_length, path_length, fd_length, tld_length, count-, count@, count?, count%, count., count=, count-http, count-https, count-www, count-digits, count-letters, count_dir, use_of_ip.

Various machine learning techniques such as Logistic Regression, Naive Bayes, SVM, Decision Tree, Random Forest, LDA, ANN(Adam), ANN(Adamax), ANN(Nadam) and AdaBoost were used. It was found that all the models had high accuracy, f1, precision and roc-auc scores on the test set. Out of all the models Random Forest performed the best with an accuracy of 99.72%.

Further there still exists class imbalance in the dataset which can be covered up using various other machine learning techniques like SMOTE.

References

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

[2] Cho Do Xuan, Hoa Dinh Nguyen and Tisenko Victor Nikolaevich, “Malicious URL Detection based on Machine Learning” International Journal of Advanced Computer Science and Applications(IJACSA), 11(1), 2020. http://dx.doi.org/10.14569/IJACSA.2020.0110119

[3] Scikit Learn, https://scikit-learn.org/stable/ (accessed Nov. 7, 2021).

[4] Keras, https://keras.io/guides/sequential_model/ (accessed Nov. 9, 2021)