International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 04 | April - 2023 Impact Factor: 8.176 ISSN: 2582-3930

Malicious URL Detection Using Machine Learning

Ralla Anand Vardhan[1], Akash Pabba[2], K Veena[3], Uttla Vishal Kumar[4], Balaraj Yadav[5]

[1,2,4,5] *Project Students,*
[3] *Project Guide*
Department of Computer Science & Engineering
Hyderabad Institute of Technology and Management. Hyderabad

Abstract - Malicious Uniform Resource Locators (URLs), or malicious websites, are one of the most common threats to web security. They host unwanted content (spam, malware, inappropriate ads, scams, etc.) Your visit to this website may have been prompted by emails, advertisements, web searches or links from other websites. Either way, the user must click on the malicious URL. The growing prevalence of phishing, spam, and malware has led to a strong need for a reliable solution that can classify and identify malicious URLs. In this paper, we address malicious URL detection as a binary classification problem and evaluate the performance of several known machine-learning classifiers.

Key Words: Malicious URLs, Machine Learning, Phishing, Spam, Malware, Fraud. Key Words: Malicious URLs, Machine Learning, Phishing, Spam, M alware, Fraud.

1.INTRODUCTION

Covid 19 has had a huge impact on the growth of online busin esses such as online banking, ecommerce, and social media. Unfortunately, with advances in technology come the most modern techniques of exploiting users. Such attacks often involve malicious websites that steal a variety of private information that hackers can exploit. In terms of malicious URL detection, traditional classification techniques such as blacklists [1], regular expressions [2], and signature matching methods [3] are challenging due to a large amount of data pattern changes over time, and complex relationships between Some It seems inevitable that malicious sites will not be blacklisted. Just as any file on your computer can be found by typing the file name, a URL can also be used to locate any web page. This is the URL of the resource on the WWW. Each URL has two main components. The first is the protocol. For the URL https://www. google.com is the HTTPS protocol identifier. Hypertext Transfer Protocol Secure (HTTPS), for loading hypertext documents. Other protocols include File Transfer Protocol (FTP), Domain Name System (DNS), and others. The second is the resource identifier.

A resource identifier is the address of a web page on the Inter net. Featured artwork in this article Consider identifying bad URLs and examine evaluation metrics of various machine learning classifiers [41. The data source is a public dataset from Kaggle [51] containing 450,000 URLs. The best classifier was used to detect malicious URLs from the open high website [6]. The rest of the document is divided into the following sections. The second section describes URL classification. Section III presents the machine learning classification techniques used to solve this problem. The visualization of the data set is presented in section IV. Section V explains the experimental results obtained. Section VI gives the conclusion.

2. PROBLEM DECRIPTION

URLs are widely used and abused to exploit user vulnerabilities. This article focuses on classifying any URL as benign or malicious. Additional comparisons of Logistic Regression (LR), Stochastic Gradient Descent (SCID), Random Forest (RF), Support Vector Machine (SVM), Naive Bayes (NB), K-plus near neighbors (KNN) and decision trees (DT). The most powerful classifier is used to detect malicious websites from Open Phish. The proposed framework is divided into five phases:

* Data collection: An annotated data set of malicious and benign websites is collected from the Kaggle repository.
* Data cleaning and extraction: pre-processing includes additional feature extraction, normalization, categorical value encoding, descent sta value (SGI), random forest (RE), support vector machine (SVM), Bayes naive (LW), K nearest neighbors (KNN) and decision trees (DT) on 80% of the data).
*Model testing and optimization: The trained model is tested on the remaining 20% of the data. Define hyperparameters to improve precision, accuracy, and recall.
* Model Comparison: Compare machine learning classification techniques based on evaluation metrics. Normalization and treatment of missing data Classification techniques Classification [7] is a machine learning process used to classify given data into a set of classes. Data can be in structured or unstructured format. The process includes pre-processing, model training, and classification of data into given classes. Classes are also called targets, categories, or labels. There are two types of classification namely binomial classification and multiclass classification. Some key areas where classification is used include classifying spam or spam, classifying tweets as negative or positive sentiment, classifying different images of fruits, animals, insects, etc., and many more complex tasks.
• Logistic Regression: Linear Regression is a linear machine learning algorithm for classification. In logistic regression, the probabilities of possible classes are calculated using a sigmoid function. The sigmoid function is used because the function varies from 0 to l. Logistic regression is used to understand

the relationship between independent and dependent variables. It is easy to implement and is the most computationally efficient algorithm compared in this article. Logistic regression can be used in case of binomial classification. It assumes that the independent variables are uncorrelated.

• Stochastic gradient descent: SGI) Stochastic approximation for optimizing gradient descent in iterative methods. Its advantages include ease of implementation. It's also less computationally expensive. Since it is a linear model, it does not handle nonlinear relationships between dependent and independent variables. It is sensitive to normalization, and requires hyperparameter tuning.

• Naive Bayes: Naive Bayes is a statistical classification model primarily based on Bayes' theorem Stochastic Image Gradient Descent It assumes a very weak correlation between the independent variables. In general, Naive Bayes classifiers are linear models, but when fed with a kernel density function, the model can classify nonlinear data very well. The main advantage of Naive Bayes is that the learning rate is higher than some more complex algorithms. It requires fewer data than other models. The downside is lower accuracy compared to other machine learning classifiers.

• K-Nearest Neighbors: K-Nearest Neighbors is a statistical classification model. Data points in this model are ranked based on the proximity of their neighbors. It is a nonparametric model. The number of neighbors is the primary hyperparameter passed to the function. Highlights include top models that provide good accuracy when trained with large data. It can also handle noisy data. Finding an optimal classification model is expensive because we have to test the model for different values of k (i.e., number of neighbors).

• Decision tree: A decision tree classifier builds a tree to classify data by generating a set of rules. The distribution of nodes in a decision tree is based on information gain and entropy. Unlike artificial neural networks, which look like black boxes, decision trees can be visualized and easily understood. Both numeric and categorical data types can be used in decision trees. Decision trees tend to overfit data when overtrained. Completely different trees may be generated due to slight changes in the data.

• Random Forest: Random Forest classifiers are a class of ensemble classifiers that satisfy many decisions Image Description Trees on different subsets of data. The final model is based on the average of different trained decision trees. It often outperforms decision trees and even solves the overfitting problem. It cannot be used for real-time applications due to the high computational cost of training a random forest classifier. It is a complex algorithm that can also be trained.

Data Visualization:

• Data Collection: An open-source dataset of 450,000 web pages was collected from a Kaggle repository for training and evaluation of machine learning models. The data consists of two properties: the URL and the tags, as shown below.

	url	label
0	https://www.google.com	benign
1	https://www.youtube.com	benign
2	https://www.facebook.com	benign
3	https://www.baidu.com	benign
4	https://www.wikipedia.org	benign

image dataset collected

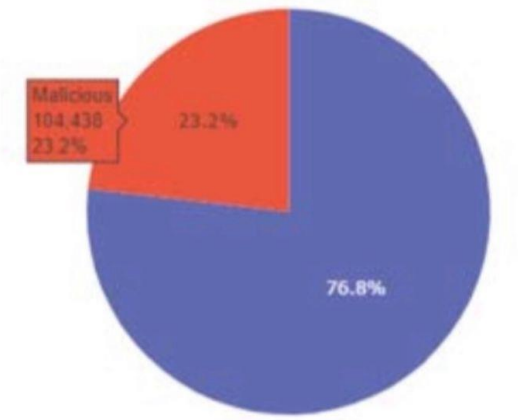

image data analysis

23.2% (104,438) of the complete data are malicious URL and the rest are benign URLs as shown in figure 10.

A set of data is visualized by grouping them. The top 20 domains grouped by domain name on a logarithmic scale are shown in Figure II. Figure 12 and 13 show the top 20 domains grouped by subdomains and suffixes on a logarithmic scale.

3.FEATURE EXTRACTION

Feature extraction is the process of representing or augmenting features that make machine learning models perform better. It helps in reducing dimensions and facilitates faster processing. The most common approaches are linear discriminant analysis and principal component analysis.

Table I shows the list of features extracted for detecting malicious websites.

Categorical features such as subdomain, domain, suffix, and target are encoded into numbers because the machine learning model cannot interpret the text directly. The coding technique used is a count coder. Malicious sites are set to I in the target column, while benign sites are set to 0.

4.CONCLUSION

A dataset of URLs has been visualized. We can conclude that the models used to achieve high prediction accuracy, but the random forest achieves the highest Fl score and accuracy. The accuracy of a trained random forest classifier on open high data can be increased by training it on more balanced data, i.e., data containing malicious and non-malicious websites in almost equal proportions. Analysis helps to find this out

REFERENCES

1. Dhanalakshmi Ranganayakulu, Chellappan C., Detecting Malicious URLs in E-mail an Implementation, AASRI Procedia, Vol. 4, 2013, Pages 125-131, ISSN 2212-6716, https://doi.org/10.1016/j.aasri.2013.10.020.
2. Yu, Fuqiang, Malicious URL Detection Algorithm based on BM Pattern Marching, International Journal of Security and Its Applications, 9, 3344, 10.14257/ijsia.2015.9.9.04.
3. K. Nirmal, B. Janet, and R. Kumar, Phishing - the threat that still exists, 2015 International Conference on Computing and Communications Technologies (local), Chennai, 2015, pp. 139-143, doi 10.1 109/1CCCT2.2015.7292734.
4. https: //openphish.COm/ accessed a 27. 01. 2021 Doyen Sahoo, Chenghao lua, Stev'en C. H. Hoi, Malicious URL Detecnon using Machine Learning: A Survey, lcs.LGl, 21 Aug 2019
5. Rakesh Verma, Avisha Das, What's in a URL' Fast Feature Extraction and Malicious URL Detection, ACM ISBN 978-145034909-3/17/03