Prediction of Malicious and Benign Websites using Supervised Learning

Anushadevi Rajkumar, ID: 220210265, ECS784P – Data Analytics - 2022/23

Abstract—The following report present a prediction model for detecting malicious and benign websites based on each website's network traffic using a dataset from Kaggle. The project explores data exploration, feature selection, training of the prediction model, performance tuning and testing.

The dataset utilized for the training of the model consists of URLs as an anonymous identification and focuses more on the details of each website's network traffic, URL characteristics and server details. The main objective of this project is to build a model that predicts based on the network traffic rather than the URL link itself.

The report entails the details on the theoretical background research done, data analysis and cleaning, methodologies used in order to obtain an optimal prediction model.

Introduction

Exploring the internet has become a daily necessity for everyone, from reading news/ articles to checking websites from time to time all age groups have equipped very well on websites. As this has become a technology, we all rely on every time, various cyber-attacks have raised drastically.

In the year 2022, 611 cyber-attacks have been found in the UK and this represents only 57% among all the incidents. Amongst these 611 attacks, the majority consisted of phishing, malware, and ransomware. These major security threats all could be caused by merely clicking onto a malicious web link, hence in order to analyze this in depth a data set of malicious and benign websites was utilized to build a machine learning model that could predict the website's safety.

Theory

Adversaries have used the Web as a vehicle to deliver malicious attacks such as phishing, spamming, and malware infection. For example, phishing typically involves sending an email seemingly from a trustworthy source to trick people to click a URL (Uniform Resource Locator) contained in the email that links to a counterfeit webpage. [1]

A URL is a unique address that specifies a website's location on the computer network, though the URL does not only specify the location but also provides details on the query being requested from the client to server.

A URL can be classified as a malicious link using its network communications and URL structure, since the aim of this project is to determine using the network traffic, we will be focusing more on the aspects of how we could determine one using its network attributes.

The following are the processes taken by the client (in most of the cases it's the browser) and the web server when a URL is clicked or entered.

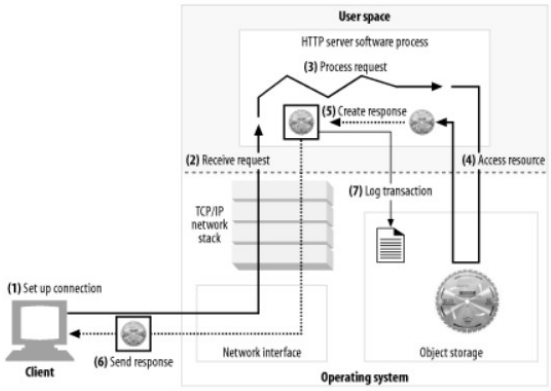


Figure 1 Client-Server Communication

1. Set up connection— establishes or closes a connection with TCP (Transmission Control Protocol) or UDP (User Datagram Protocol) in order to proceed or end a request to view a site or download the data from the internet. This connection is vital for the client as it gets the real address from the URL through a DNS server.

2. Receive request—reads an HTTP request message from the network which consists of the request line, header and body.
    1. Request Line: includes the HTTP method (the action of the request (such as GET, POST or PUT), the requested resource (URL) and the HTTP's version.
    2. Request Header: consists of additional information on the request which usually include the user agent details (the web browser that is creating the request).

3. Request Body: consists of additional data such as form data sent to the user space. This is usually an optional component while receiving the request.

3. Process request—interprets the request message and takes action accordingly.
4. Access resource—accesses the resource specified in the message.
5. Construct response—creates the HTTP response message with the right headers which are specified on the details of the request header.
6. Send response—sends the response back to the client with the requested information.
7. Log transaction—place notes about the completed transaction in a log file. [2]

One of the main aspects of this process is when the request sent from the client to the server and the response sent from the server to the client passes through the network interface (TCP/ IP network stack).

The TCP/ IP network stack consists of the foundational protocols used by the internet, where TCP is responsible for handling connection between the client and server and IP handles the packet routing among different network devices. Packets are the units of data sent from a client to server system and vice-versa, it is constructed with information of source and destination addresses, sequence number, acknowledgement number and data payload. A packet size may vary depending on several factors such as application and network requirements, but a typical size of a packet is around 64 to 1500 bytes, and a packet size from a malicious website could either strangely be too small or too large.

Throughout this report, elements from the dataset are analysed in order to train the model to predict a safe and unsafe website using supervised learning. Supervised learning is used in this as the feature variables are labelled and classified as seen from the dataset.

Data processing

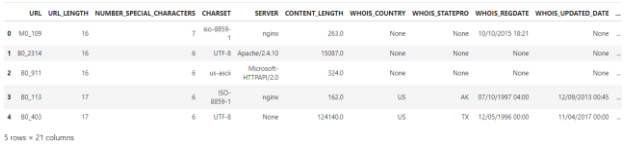
To train our model, we import the dataset chosen from Kaggle – Malicious and Benign Dataset. [3]

The dataset consists in total of 21 columns and 1781 rows, in table 1 is the data description of what each column represents in which we'd visualize and analyze those with the prediction variable. In order to categorize the variables into numerical and categorical variables, types of each variable are determined. A numerical variable represents quantitative values where in a categorical variable is a qualitive value which are not numerical in nature.

TABLE I. ADDITIONAL PREPROCESSING AND THEIR IMPACT AFTER CROSS VALIDATION

Variable	Description	Type
URL	anonymous identification of the URL	Object
URL_LENGTH	number of characters in the URL	Integer
NUMBER_SPECIAL_CHARACTERS	number of special characters present in the URL	Integer
CHARSET	character encoding standard used	Object
SERVER	operating System (OS) of the server	Object
CONTENT_LENGTH	HTTP Header Content Size	Float
WHOIS_COUNTRY	server's response location - country	Object
WHOIS_STATEPRO	server's response location - state	Object
WHOIS_REGDATE	server registration date (DD/MM/YYY HH:MM)	Object
WHOIS_UPDATED_DATE	server's latest update date (DD/MM/YYY HH:MM)	Object
TCP_CONVERSATION_EXCHANGE	number of TCP packets exchanged between server and client	Integer
DIST_REMOTE_TCP_PORT	number of ports detected that are different from the TCP	Integer
REMOTE_IPS	total number of IP addresses connected to the client	Integer
APP_BYTES	number of transferred bytes from the server to the client	Integer
SOURCE_APP_PACKETS	number of packets sent from the client	Integer
REMOTE_APP_PACKETS	number of packets recieved from the server	Integer

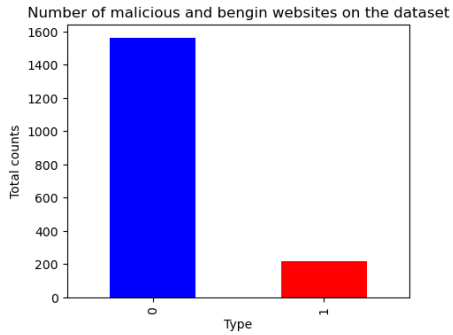
APP_PACKETS	Total number of IP packets during the client-server communication	Integer
DNS_QUERY_TIMES	number of DNS packets created during the client-server communication	Float
TYPE	web page type (1 - malicious, 0 - benign)	Integer



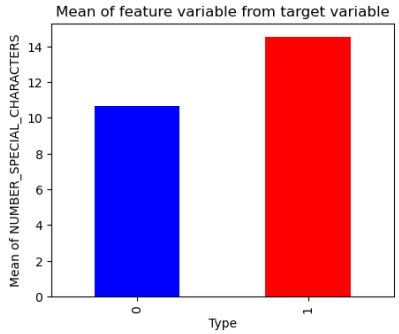
	URL	URL_LENGTH	NUMBER_SPECIAL_CHARACTERS	CHARSET	SERVER	CONTENT_LENGTH	WHOIS_COUNTRY	WHOIS_STATEPRO	WHOIS_REGDATE	WHOIS_UPDATED_DATE	...
0	M0_109	16	7	iso-8859-1	nginx	263.0	None	None	10/10/2015 18:21	None	...
1	B0_2314	16	6	UTF-8	Apache/2.4.10	15087.0	None	None	None	None	...
2	B0_911	16	6	us-ascii	Microsoft-HTTPAPI/2.0	324.0	None	None	None	None	...
3	B0_113	17	6	ISO-8859-1	nginx	162.0	US	AK	07/10/1997 04:00	12/09/2013 00:45	...
4	B0_403	17	6	UTF-8	None	124140.0	US	TX	12/05/1996 00:00	11/04/2017 00:00	...

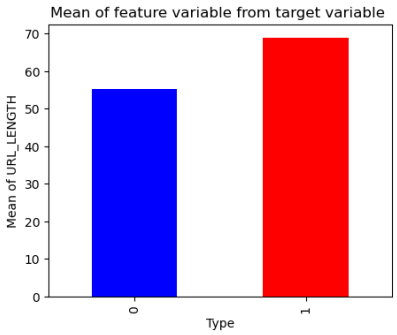
5 rows × 21 columns

Since "URL" is a non-numeric variable that is used only as a ID, it is dropped from the dataset. With 20 columns in total now, the data is further analysed. The "Type" column consists with 1 for the website being malicious and 0 for a benign website (safe website), with this data on the spotlight we compare other columns in terms of similarities, uniqueness and distributions. The dataset consists of a total of 216 malicious URLs and 1565 benign URLs.

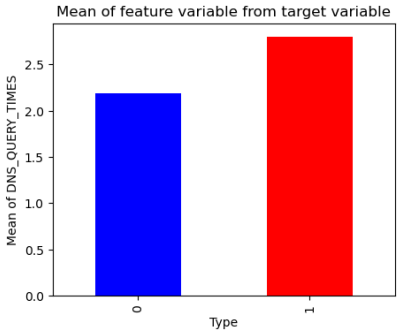


A URL could be malicious when it contains many special characters in it, we could also determine it by analysing the URL's length, though it cannot be considered in all cases.





After analysing the other numerical variables, it could be seen from the graphs below on the basis of network traffic, that the variables related to packet sizes moreover balanced for malicious and non-malicious URLs. The DNS_QUERY_TIMES variable tends to reflect more on the malicious URL data available. The number of DNS packets created could also depend on the factors of the number of resources the site is fetching, the browser and website performance. Hence, these numerical categories are vital in order to train the model.



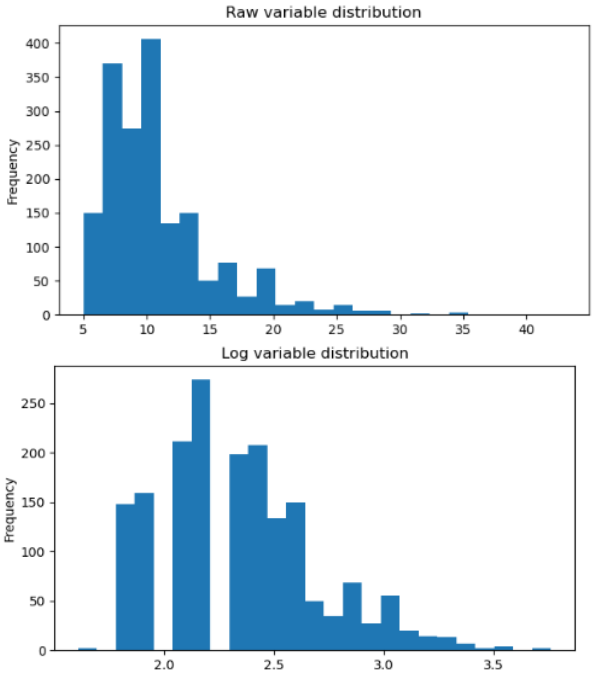
Data Pre-processing

One mandatory step that needs to be taken while dealing with datasets are to discover null values and fill them with 0.

From analysing the categorical variables, it could be noted that the WHOIS_REGDATE, WHOIS_UPDATED_DATE are timestamps, and also contains "None", these are converted into a string (format: %H:%M:%S) in order for it to be encoded as a label for training. The WHOIS_COUNTRY contains codes of countries that are duplicated such as: the United Kingdom, GB, UK, US, us, SE, se, ru and RU. These could be combined into its corresponding country code in order for the column to contain only unique values.

Another mandatory step that needs to be implemented is to normalize the data for each numerical variable. Since the numerical data are highly skewed, it needs to be in a consistent range and distributed. All the data are normalized using logarithmic transformation to create a standardized and consistent input for the project.

$$x' = \log(x)$$

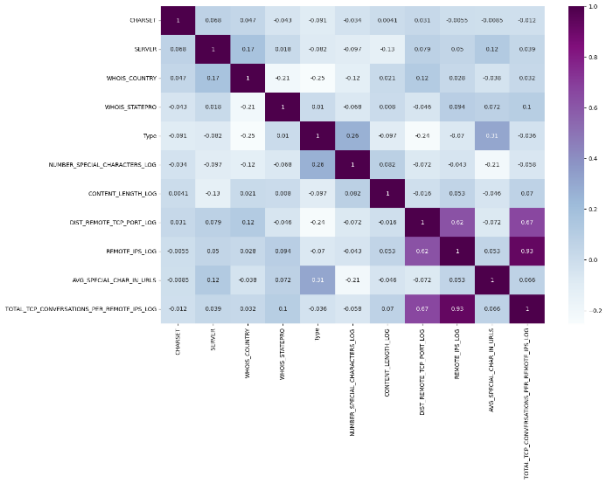


As the input data is normalized, it can now be further analysed if there are any composite variables that can be created. From the data on hand, we can further calculate the average special characters in the URL for a better prediction rate - df['NUMBER_SPECIAL_CHARACTERS'] / df['URL_LENGTH']. The total TCP conversations per remote IPS could also be determined. - df['TCP_CONVERSATION_EXCHANGE'] * df['REMOTE_IPS']

Once these new composite variables are added onto the dataset, the categorical variables can be converted into integers by categorizing each column's unique strings into numbers. This is done since the machine learning algorithms utilized in this project works well with numerical models.

Feature Selection

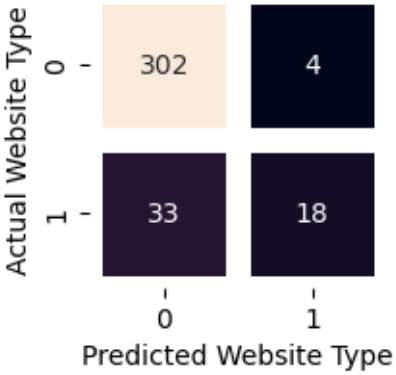
After pre-processing the data, by analyzing the data with heat map, several features were dropped due to high correlation. The features that were dropped included the initial numerical features (the features that were used to log transform) and also the features that were used to calculate the composite variable was removed from the data.



Learning Methods

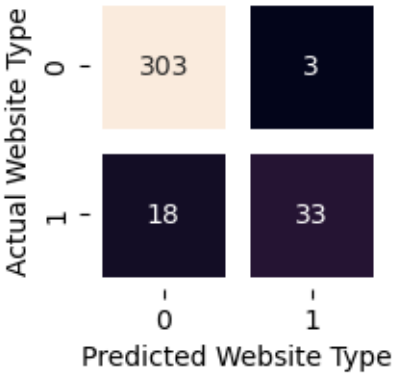
To train the model, the dataset after the feature selection was utilized. It was divided into 2 variables X and y, X consisting of the feature variables and y consisting of the target variable – Type. The dataset (X, y) is then further divided for training and testing purposes. Using sklearn library's train_test_split, the data is split with a testing data of 20% and training data of 80%.

From analyzing the dataset, we know that our target values are 0 and 1 hence making it a classification machine learning process. To train the model to classify malicious and benign URLs, Logistic Regression is used as the first learning algorithm. This was chosen since the target variables are 0 and 1. Logistic regression is used to understand the relationship between independent and dependent variables. [4] With training the data in this model, we receive an accuracy of 90.3%.



The second learning algorithm utilized in order to for training the model is the K-nearest neighbor (KNN). This learning method was opted due to training with a dataset where there is no linear relation between the feature and target variable. KNN algorithm is an instance based classifier unlike the logistic regression, where instead of learning a model, it compares each data with past data instances. By training the data under KNN

algorithm, the accuracy received with the training data is 95.9%.



Analysis and Testing

During the testing phase of this project, various feature variables (apart from the initial variable used to determine the logarithmic transformation) were not dropped. Training with these the accuracy on the training model was at 80%. Hence, it could be said that it is vital to remove the feature variables that are highly correlated.

On the testing phase of the learning models, k-fold cross-validation was opted with 5 folds. For both the learning models, the mean of cross-validation was satisfactory – 89.8% for logistic regression and 94.3% accuracy on KNN algorithm.

Initially as the feature selection was done manually by analysing bi-variant variables and creating composite variables, an alternative way of feature reduction was utilized. Principal Component Analysis (PCA) is a technique used to reduce features for high-dimensionality datasets. The technique is applied by using the original dataset after pre-processing; hence the shape of this dataset is (1781 rows, 19 columns). From the graph below, we can see that the cumulative explained variance stabilizes when it reaches a total of 8 components. By choosing these 8 components, we again apply both the learning algorithms. To gain the accuracy from both these models, we run the model with the training and test data; for Logistic Regression the accuracy of training was 90% and testing was 88.2%. As for the KNN algorithm the training accuracy was 93.7% with a testing accuracy of 93.3%. In conclusion these prediction accuracy scores are well achieved and we can see that it is slightly lower than the accuracy obtained from manual feature selected data set. One of the causes of this could be the composite variables we have added during the feature engineering process of this dataset as it represents variables with relations and increases the chances of precise predictions.

Conclusion

Detection of malicious websites could be complex and with day to day raise in cyber-crimes, these becomes a complex data to analyse. Prediction by using only the URL characteristics is getting difficult as well, as these days intruders replicate URLs nearly similar to the legitimate domain names we see on everyday basis. Hence, this project aims in predicting the malicious and benign websites through network communications of these URLs, as network traffic of these malicious sites always gives away a pattern with packet traffic spikes.

Various researches and preventions has been held in order to prevent from falling into these sorts of cyber attacks, but it is always a good practice to not open URLs from untrusted sites or emails.

To conclude this, from the results obtained by analysing and testing, the aim for predicting malicious and benign sites is achieved by data pre-processing, feature selection and training the model with two learning methodologies – Logistic Regression and KNN algorithm. The model could also be trained with more complex algorithms such as Neural Networks, Naïve Bayes and Ensemble Learning. From the literature review conducted for this project various journal papers contribute to training with complex algorithms for better error prone model and prediction - Cyber Threat Intelligence-Based Malicious URL Detection Model Using Ensemble Learning [5] and A Malicious URL Detection Model Based on Convolutional Neural Network [6].

References

[1] C. B. Z. B. Hyunsang, “Detecting malicious web links and identifying their attack types,” 2023.

[2] B. T. M. S. A. A. S. R. David Gourley, HTTP: The Definitive Guide.

[3] M. L. C. t. D. M. Websites., Urcuqui, C., Navarro, A., Osorio, J., & Garcıa, M., CEUR Workshop Proceedings, 2017.

[4] J. B. J. A. K. R. Shantanu, “Malicious URL Detection: A Comparative Study,” 2021.

[5] M. A. F. S. J. A. a. M. A. Fuad A. Ghaleb, “Cyber Threat Intelligence-Based Malicious

URL Detection Model Using Ensemble Learning," in *Sensors*, 2022.

[6] X. R. S. L. B. W. J. Z. a. T. Y. Zhiqiang Wang, "A Malicious URL Detection Model Based on Convolutional Neural Network," in *Hindawi*, 2021.