Introduction

In today's interconnected world, where technology permeates every aspect of our lives, cybersecurity is vital to prevent financial losses, reputational damage, and disruption of services. By implementing robust cybersecurity measures, individuals, businesses and organizations can mitigate risks and defend against evolving cyber threats, ensuring a safe and secure digital environment for all. To implement a robust cybersecurity system, we need AI to mitigate following risks:

- Rising cyber threats
- Evolution of attack techniques
- Complexity of Malware
- Human error and social engineering

Problem Statement

Malicious URLs are commonly used by cybercriminals to initiate various forms of cyber-attacks, including malware, distribution, phishing and ransomware. Detecting and blocking these URLs will prevent these attacks from being successful, protecting both individuals and organizations. The objective of this project is to classify malicious URLs based many factors such as:

- Domain Information
- Path and Query parameters
- SSL Certificate details
- Host reputation
- Network features

This is a **binary classification problem** to identify whether a URL is malicious or not

About the Data

```
Number of rows:         6728848
Number of columns:      60
```

The Dataset was retrieved from Kaggle: Click Here. Dataset consists of 60 columns and more than 6.5 million rows.

Due to system limitations, only 500,000 records were considered for model prediction. While creating a smaller dataset, we need to address the problem of class imbalance. Hence, 250,000 records from each class were considered for model training and evaluation.

Exploratory Data Analysis

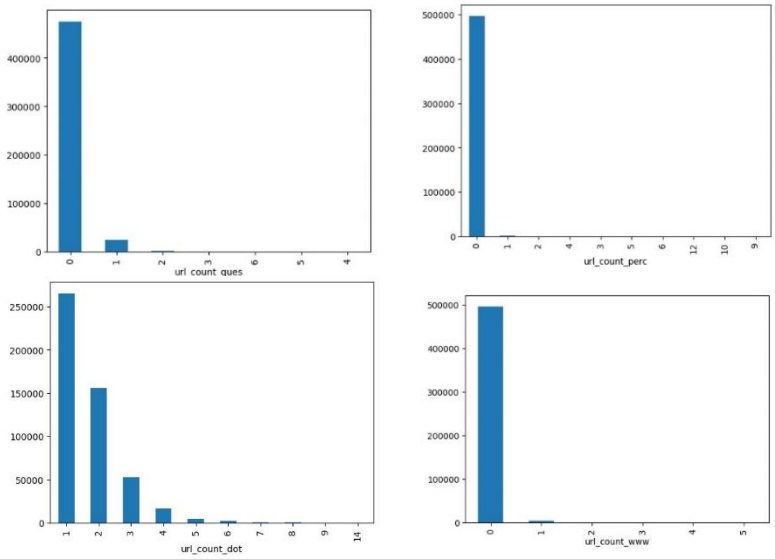
- **Checking the Null values within each column:**
  Fortunately, none of the columns had any null values in it.

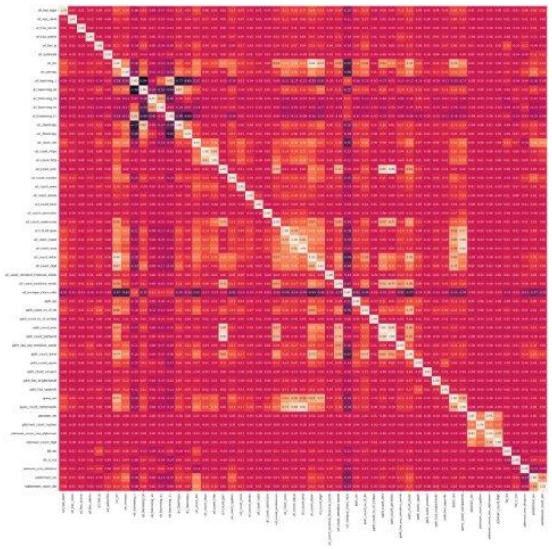
```
Column Name             Count of null values
url                     0
label                   0
source                  0
url_has_login                   0
url_has_client                  0
url_has_server                  0
url_has_admin                   0
url_has_ip                      0
url_isshorted                   0
```

url	source	tld
cpuggsukabumi.id	majestic_million	id
members.tripod.co...	data_clean_test_m...	com
topoz.com.pl	dmoz_harvard	com.pl

- **Categorical Columns:** None of the categorical columns were significant for model training. The URL column just tells the URL, the source column just tells us the source of information about the URL features. The tld column just tell us the dot extension of the URL.

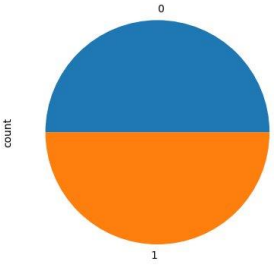
- **Univariate Analysis – Numerical columns:** After visualizing the plots, it was found the many categorical columns were encoded into their numerical format. Most columns were highly skewed or belongs to single category. Skewed columns or columns belonging to a single category in machine learning can lead to biased predictions. If the categorical column contains mostly a single category, it won't provide any meaningful information for the training our machine learning model. Skewed distributions also violate the assumptions of many statistical methods. Models, such as logistic regression, assume the data follows a normal distribution. Hence, many columns were removed from the training processes

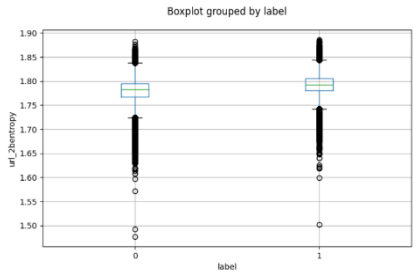




- **Multi-variate Analysis – Numerical columns:** High correlation among independent variables (also known as multicollinearity) in a dataset can cause several issues during model training. Hence, we need to remove one of columns that are highly collinear with another column. Out of initial 60 columns, only selected columns were considered for model training post univariate and multivariate analysis as most columns were either skewed/ belonging to a single category in a column or highly correlated with other independent features.

- **Target Variable:** Our data is balanced having equal records for each class within the target. Here, value 0 represents URLs that are not malicious and 1 denotes the URLs that are dangerous

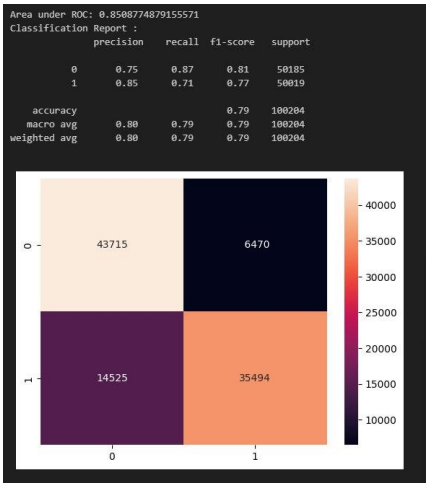




- **Bi-variate Analysis:** Tries to find the patterns among the independent variables with the target variable (Malicious URL or not). Many questions were answered such as does URL length suggests anything about the URL being malicious, are there any patterns between URL entropy and the URL being malicious, etc.

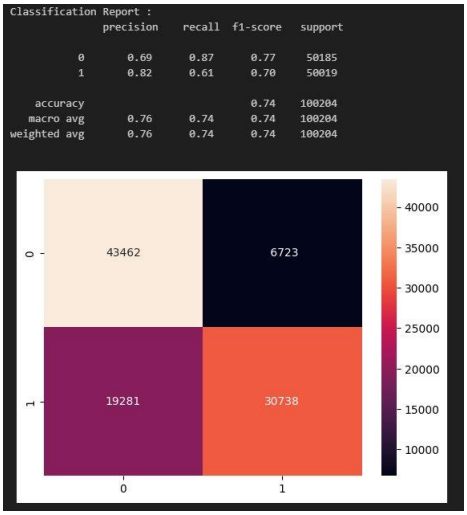
Model Building and Training

For model training, pyspark was leveraged. The dataset was randomly split for training and testing.



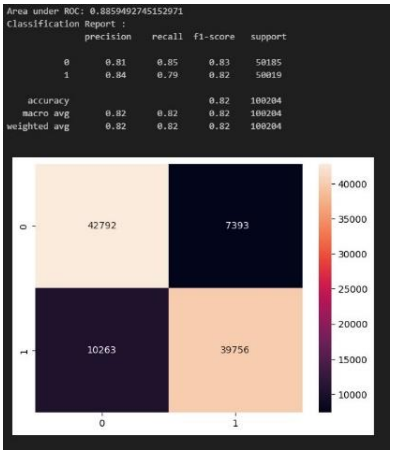
- **Logistic Regression**

Logistic regression was considered for its simplicity. Logistic regression can be easily interpreted. Initially, the features were converted into vectors using Vector assembler. The dataset was then randomly split for training and testing. 80% of the dataset was used for training (~400k records) and 20% of the dataset was used for testing (~100 k records). Area under ROC, classification report and confusion matrix was used to evaluate the logistic regression model. According to the model evaluation, Logistic regression model gives an accuracy of 79%.



- **Naive Bayes**

Naive Bayes is computationally efficient, making it suitable for large datasets. Initially, the features were converted into vectors using Vector assembler. The dataset was then randomly split for training and testing. 80% of the dataset was used for training (~400k records) and 20% of the dataset was used for testing (~100 k records). Area under ROC, classification report and confusion matrix was used to evaluate the logistic regression model. According to the model evaluation, Naive Bayes model gives an accuracy of 74%.

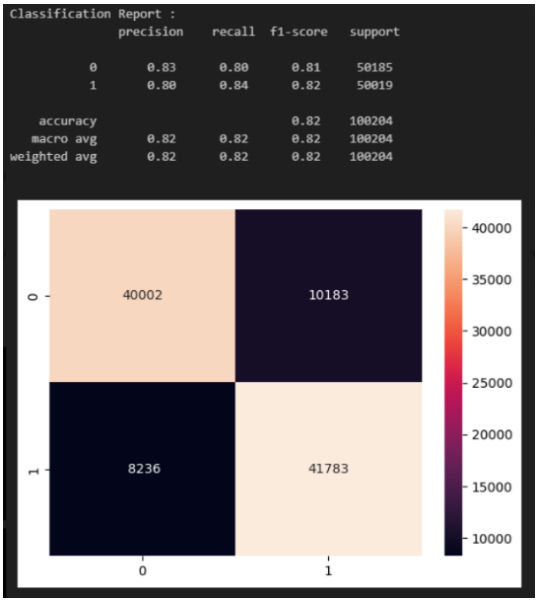


- **Random Forest**

Although Random Forest is a black-box model, it usually performs better due to its ability to generate multiple trees and pooling the decisions made on individual trees. Initially, the features were converted into vectors using Vector assembler. The dataset was then randomly split for training and testing. 80% of the dataset was used for training (~400k records) and 20% of the dataset was used for testing (~100 k records). Cross-validation was further utilized during model training for hyperparameter tuning and mitigate overfitting. Area under ROC, classification

report and confusion matrix was used to evaluate the Random Forest model. According to the model evaluation, Random Forest model gives an accuracy of 82%.

All the confusion matrix above shows that the false negatives are more than the false positives and as per our problem statement handling false negatives (Predicting malicious URLs as non-malicious) are far more critical than false positives. Hence, we need to change the threshold of the predicted values to decrease the number of false negatives while maintaining similar accuracy.



- **<u>Finetuning the best model (Random Forest)</u>**

Out of all the models developed Random Forest performs the best. We need to finetune the model as we need to decrease the amount of false negative (Predicting malicious URLs as non-malicious) more than false positives while maintaining similar accuracy. Initially, the features were converted into vectors using Vector assembler. The dataset was then randomly split for training and testing. 80% of the dataset was used for training (~400k records) and 20% of the dataset was used for testing (~100 k records). Cross-validation was further utilized during model training for hyperparameter tuning and mitigate overfitting. Classification report and confusion matrix was used to evaluate the Random Forest model. According to the model evaluation, Random Forest model gives an accuracy of 82%. Although, the accuracy of the model remains the same by manipulating the threshold value, we decreased the number of false negatives which is critical for the problem statement

<u>Conclusion</u>

In Conclusion, our project was aimed to address the binary classification problem of Malicious URL detection, evaluating three distinct models: Logistic Regression, Naïve Bayes and Random Forest. Among these, Random Forest emerged as the top-performing model, achieving an impressive accuracy of 82%. The significance of this project lies in its practical implications for enhancing online security measures. By leveraging advanced machine learning techniques such as Random Forest, we can effectively combat cyber threats posed by malicious URLs, thereby safeguarding users from potential harm such as phishing attacks, malware infections and data breaches.