



# Detection and categorization of Malicious URL's

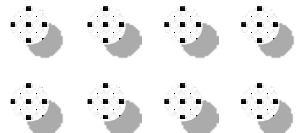
Url Prober Team  
05 April 2022

# Agenda

- Introduction
- Challenges
- Solutions
- Models and Results
- Conclusion and Future Study



Illustrations by Pixeltrue on [icons8](#)



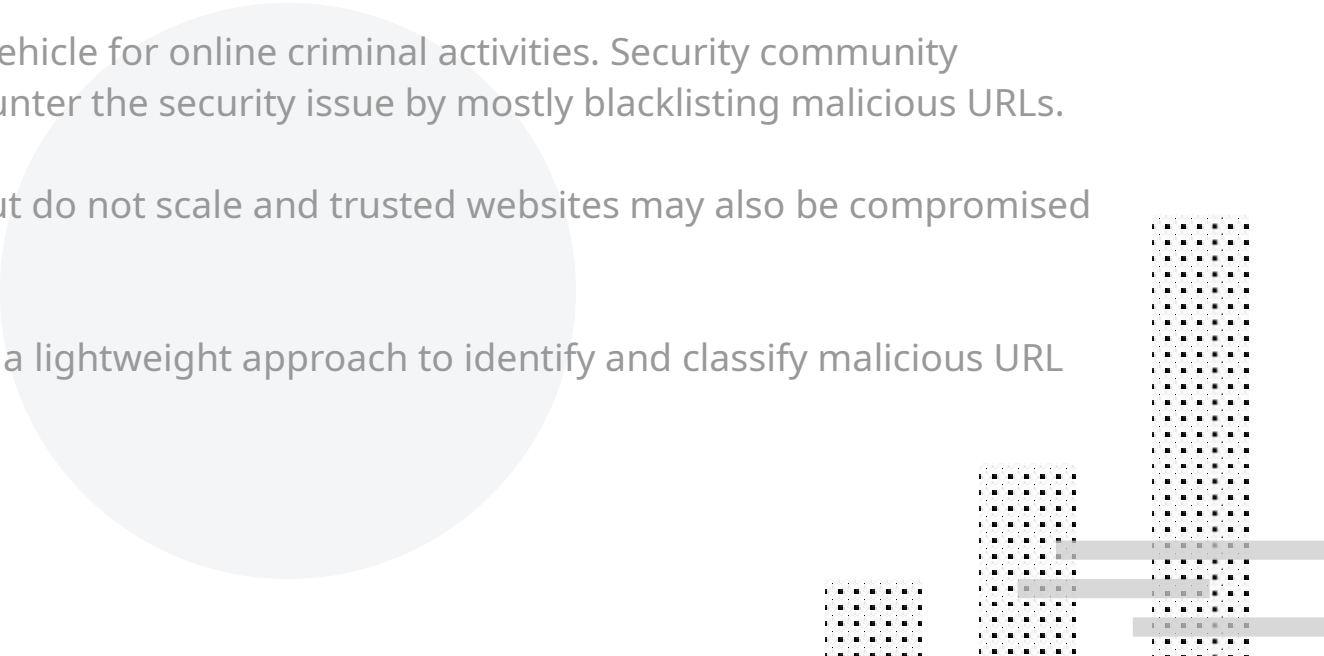


# Introduction

URLs are used as the main vehicle for online criminal activities. Security community developed techniques to counter the security issue by mostly blacklisting malicious URLs.

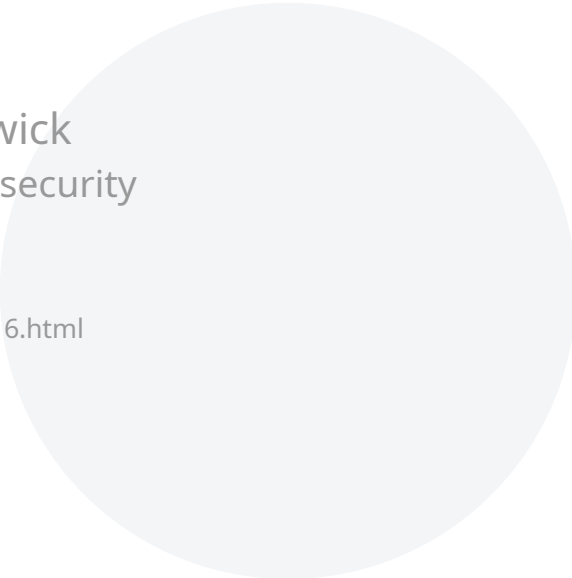
That approach works well but do not scale and trusted websites may also be compromised via defacement of URL.

This study will try to explore a lightweight approach to identify and classify malicious URL using machine learning.





# Dataset



University of New Brunswick  
Canadian Institute for Cybersecurity

URL dataset (ISCX-URL2016)  
<https://www.unb.ca/cic/datasets/url-2016.html>



# The five different types of malicious URLs

## Benign URLs

Over 35,300 benign URLs were collected from Alexa top websites. The domains have been passed through a Heritrix web crawler to extract the URLs. Around half a million unique URLs are crawled initially and then passed to remove duplicate and domain only URLs. Later the extracted URLs have been checked through Virustotal to filter the benign URLs.

## Spam URLs

Around 12,000 spam URLs were collected from the publicly available WEBSPAM-UK2007 dataset.

## Phishing URLs

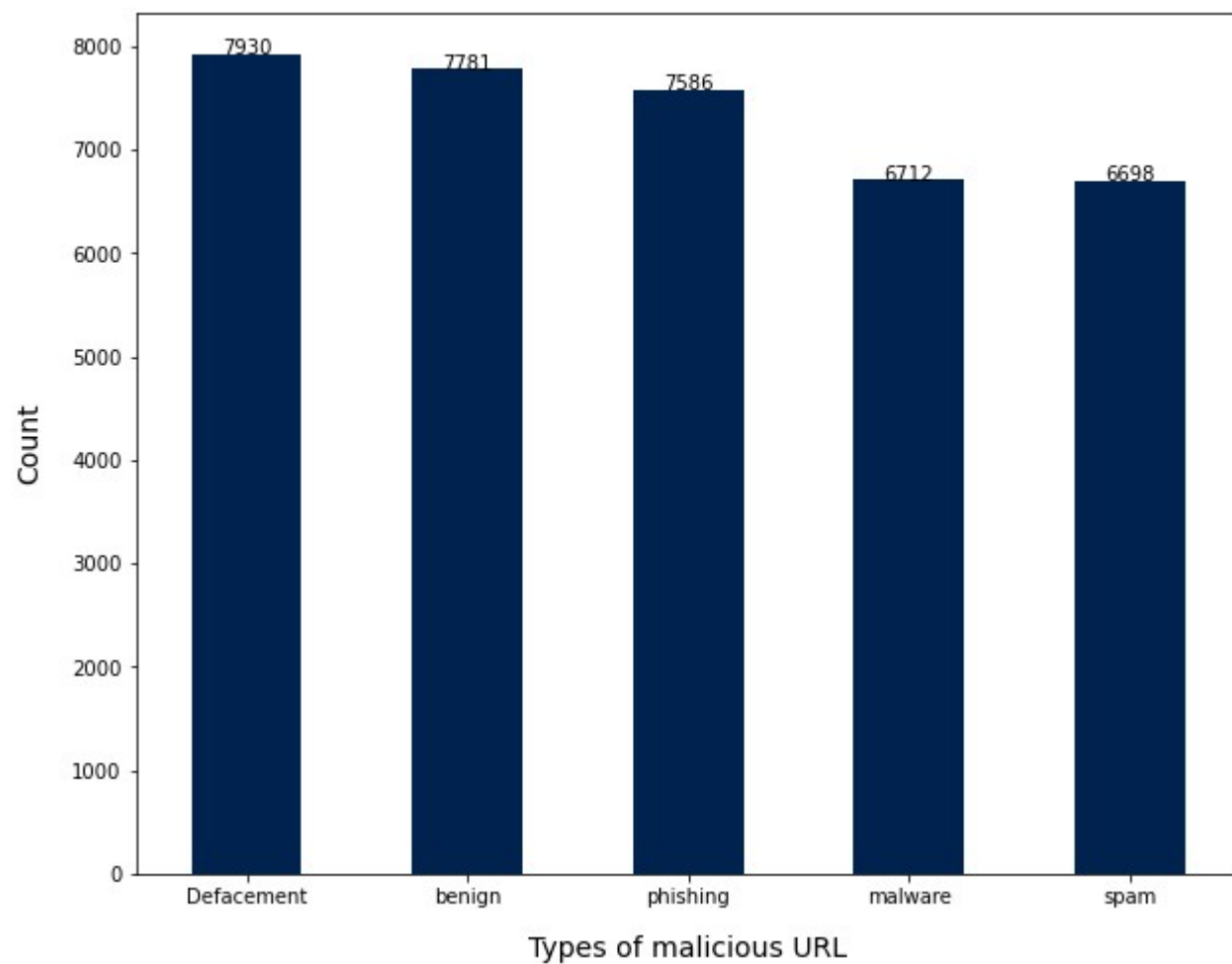
Around 10,000 phishing URLs were taken from OpenPhish which is a repository of active phishing sites.

## Malware URLs

More than 11,500 URLs related to malware websites were obtained from DNS-BH which is a project that maintain list of malware sites.

## Defacement URLs

More than 45,450 URLs belong to Defacement URL category. They are Alexa ranked trusted websites hosting fraudulent or hidden URL that contains both malicious web pages.



# Challenges



The dataset contains Null and NaN values.



Lack of description on individual features.



Difficult to determine the correlations because of large number of features.

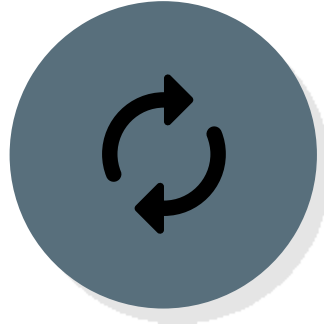
# Solutions



The dataset contains Null and NaN values.



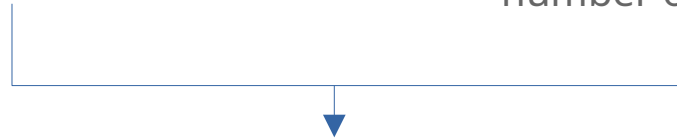
Replace with mean or drop the feature.



Lack of description on individual features.



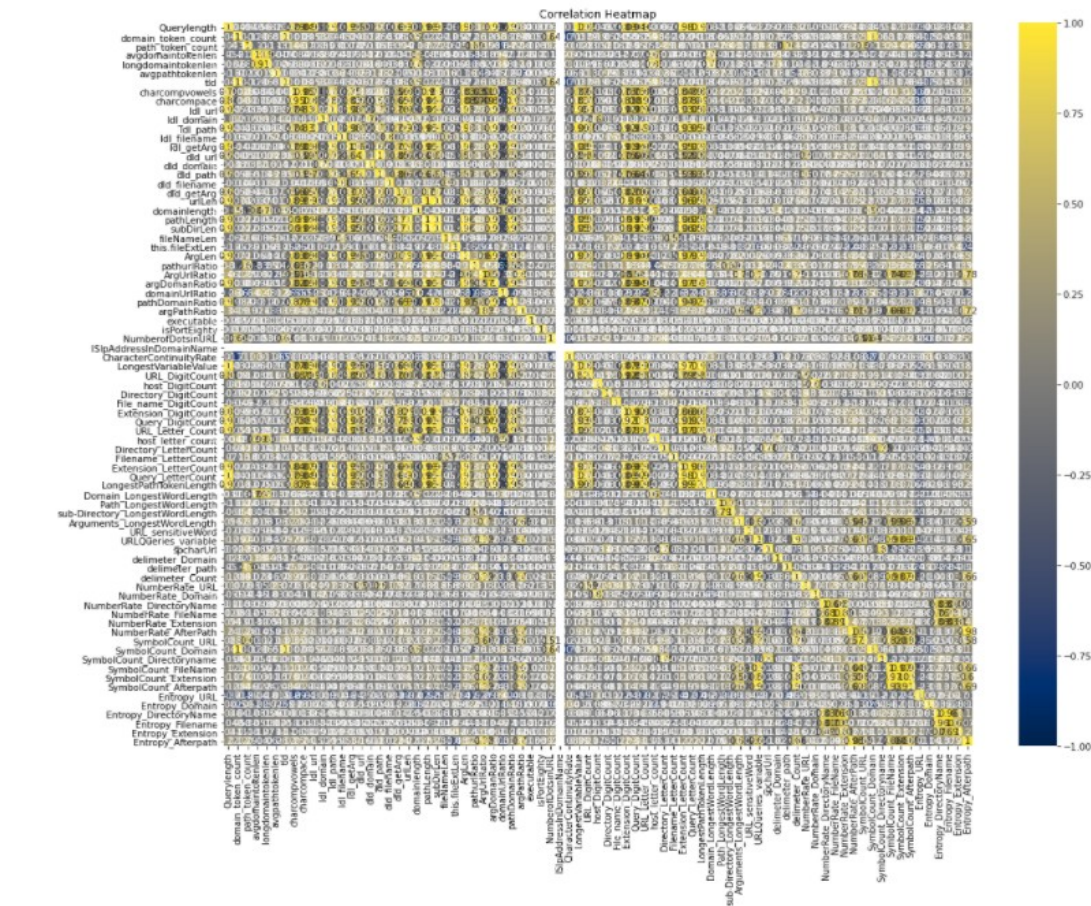
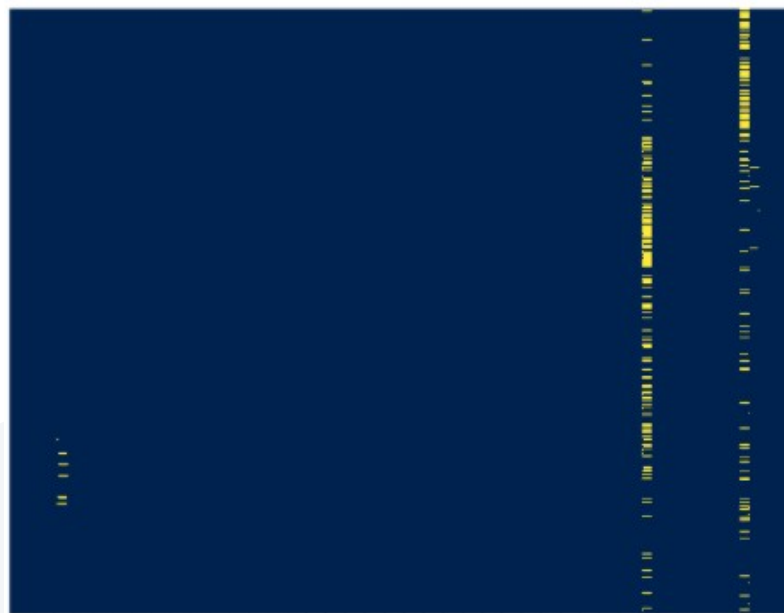
Difficult to determine the correlations because of large number of features.



Feature Selection



# Challenges



# Data Cleaning and Preparation

```
help(loader.prepare_data)
```

Help on method prepare\_data in module loader\_nb:

prepare\_data(data, fill\_na=True, feature\_selection=True, show\_graph=False) method of loader\_nb.UrlDatasetLoader instance  
(DataFrame, boolean, boolean) --> X and y of the dataframe.

This function returns the X and y of the malicious url dataframe.

Parameters  
-----

fill\_na : True to fill the na records with mean values otherwise drop the features.

feature\_selection : True to remove one or more features that have a correlation higher than 0.9 otherwise do not perform that type of feature selection.  
<https://towardsdatascience.com/feature-selection-correlation-and-p-value-da8921bfb3cf>

show\_graph : True to display the graph after applying fill\_na or feature\_selection.

# Data Cleaning and Preparation

```
help(loader.prepare_data)
```

Help on method prepare\_data in module loader\_nb:

prepare\_data(data, fill\_na=True, feature\_selection=True, show\_graph=False) method of loader\_nb.UrlDatasetLoader instance  
(DataFrame, boolean, boolean) --> X and y of the dataframe.

This function returns the X and y of the malicious url dataframe.

Parameters  
-----

fill\_na : True to fill the na records with mean values otherwise drop the features.

feature\_selection : True to remove one or more features that have a correlation higher than 0.9 otherwise do not perform that type of feature selection.  
<https://towardsdatascience.com/feature-selection-correlation-and-p-value-da8921bfb3cf>

show\_graph : True to display the graph after applying fill\_na or feature\_selection.



This study conducts various experiments based on different combinations of fill\_na and feature\_selection when preparing the data.

# Solutions

QueryLength  
path\_token\_count  
longdomaintokenlen  
td  
charcompact  
ldl\_domain  
ldl\_filename  
did\_url  
did\_path  
did\_getArg  
domainlength  
subDirLen  
this fileExtLen  
pathurlRatio  
argDomainRatio  
pathDomainRatio  
isPortEighty  
ISIPAddressInDomainName  
LongestVariableValue  
host\_DigitCount  
File\_name\_DigitCount  
Query\_DigitCount  
host\_letter\_count  
Filename\_LetterCount  
Query\_LetterCount  
Domain\_LongestWordLength  
sub-Directory\_LongestWordLength  
URL\_sensitiveWord  
specialchars  
delimiter\_path  
NumberRate\_URL  
NumberRate\_DirectoryName  
NumberRate\_Extension  
SymbolCount\_URL  
SymbolCount\_Directoryname  
SymbolCount\_Extension  
Entropy\_URL  
Entropy\_DirectoryName  
Entropy\_Extension  
URL\_type\_obf\_type



# Solutions

Correlation Heatmap

The heatmap displays the correlation between 45 features. The features are listed on both the x and y axes. The color scale ranges from -1.00 (dark blue) to 1.00 (yellow). The diagonal is white, indicating a correlation of 1.00. The heatmap shows various positive and negative correlations between the features.

Features (X-axis):

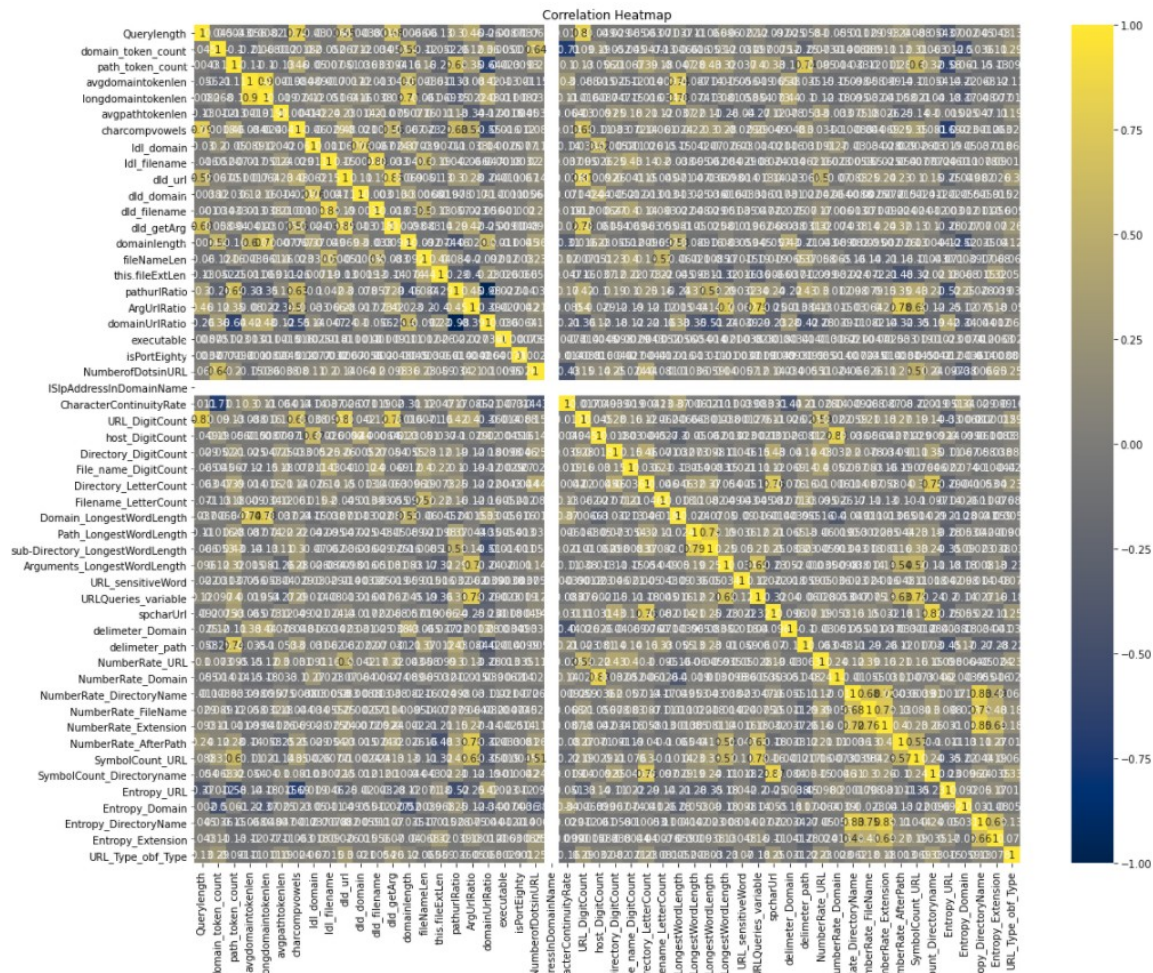
- QueryLength
- domain\_token\_count
- path\_token\_count
- avgdomaintokenlen
- longdomaintokenlen
- avgpathtokenlen
- charcomponents
- ldl\_domain
- ldl\_filename
- did\_url
- did\_domain
- did\_filename
- did\_getArg
- domainlength
- fileNameLen
- this fileExtLen
- pathUriRatio
- ArgUriRatio
- domainUriRatio
- executable
- isPortEighty
- NumberOfDotsInURL
- ISIPAddressInDomainName
- CharacterContinuityRate
- URL\_DigitCount
- host\_DigitCount
- Directory\_DigitCount
- File\_name\_DigitCount
- Directory\_LetterCount
- Filename\_LetterCount
- Domain\_LongestWordLength
- Path\_LongestWordLength
- sub-Directory\_LongestWordLength
- Arguments\_LongestWordLength
- URL\_sensitiveWord
- URLQueries\_variable
- spharUrl
- delimiter\_Domain
- delimiter\_path
- NumberRate\_URL
- NumberRate\_Domain
- NumberRate\_DirectoryName
- NumberRate\_FileName
- NumberRate\_Extension
- NumberRate\_AfterPath
- SymbolCount\_URL
- SymbolCount\_Directoryname
- Entropy\_URL
- Entropy\_Domain
- Entropy\_DirectoryName
- Entropy\_Extension
- URL\_Type\_obf\_Type

Features (Y-axis):

- QueryLength
- domain\_token\_count
- path\_token\_count
- avgdomaintokenlen
- longdomaintokenlen
- avgpathtokenlen
- charcomponents
- ldl\_domain
- ldl\_filename
- did\_url
- did\_domain
- did\_filename
- did\_getArg
- domainlength
- fileNameLen
- this fileExtLen
- pathUriRatio
- ArgUriRatio
- domainUriRatio
- executable
- isPortEighty
- NumberOfDotsInURL
- ISIPAddressInDomainName
- CharacterContinuityRate
- URL\_DigitCount
- host\_DigitCount
- Directory\_DigitCount
- File\_name\_DigitCount
- Directory\_LetterCount
- Filename\_LetterCount
- Domain\_LongestWordLength
- Path\_LongestWordLength
- sub-Directory\_LongestWordLength
- Arguments\_LongestWordLength
- URL\_sensitiveWord
- URLQueries\_variable
- spharUrl
- delimiter\_Domain
- delimiter\_path
- NumberRate\_URL
- NumberRate\_Domain
- NumberRate\_DirectoryName
- NumberRate\_FileName
- NumberRate\_Extension
- NumberRate\_AfterPath
- SymbolCount\_URL
- SymbolCount\_Directoryname
- Entropy\_URL
- Entropy\_Domain
- Entropy\_DirectoryName
- Entropy\_Extension
- URL\_Type\_obf\_Type

Color Scale:

- 1.00
- 0.75
- 0.50
- 0.25
- 0.00
- 0.25
- 0.50
- 0.75
- 1.00





# Challenge



## Finding Outliers?



**Solution**



# Isolation Forest

Unsupervised Anomaly Detection







# Isolation Forest



## Isolation:

The term isolation means 'separating an instance from the rest of the instances'. Since anomalies are 'few and different' and therefore they are more susceptible to isolation.







# Isolation Forest



## Isolation:

The term isolation means 'separating an instance from the rest of the instances'. Since anomalies are 'few and different' and therefore they are more susceptible to isolation.

## Benefits:

Unsupervised algorithm and therefore it does not need labels to identify outlier/anomaly.

Computationally efficient and low memory requirement.





# Isolation Forest



## Isolation:

The term isolation means 'separating an instance from the rest of the instances'. Since anomalies are 'few and different' and therefore they are more susceptible to isolation.


## Benefits:

Unsupervised algorithm and therefore it does not need labels to identify outlier/anomaly.

Computationally efficient and low memory requirement.

## Drawbacks:

Requires to pick the percentage of anomalies in the dataset hence we need to have at least an idea of the proportion of anomalies in our data.



# Isolation Forest

## Isolation:

The term isolation means 'separating an instance from the rest of the instances'. Since anomalies are 'few and different' and therefore they are more susceptible to isolation.

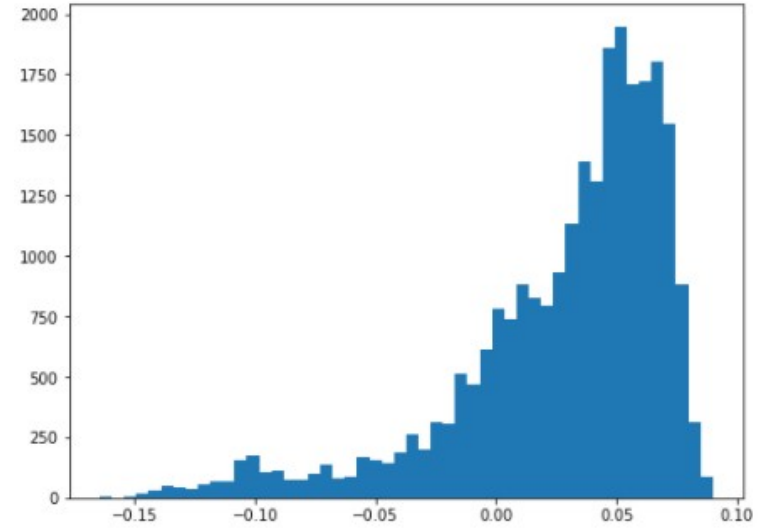
## Benefits:

Unsupervised algorithm and therefore it does not need labels to identify outlier/anomaly.

Computationally efficient and low memory requirement.

## Drawbacks:

Requires to pick the percentage of anomalies in the dataset hence we need to have at least an idea of the proportion of anomalies in our data.



# Isolation Forest

```
help(loader.perform_anomaly_detection)
```

Help on method perform\_anomaly\_detection in module loader\_nb:

perform\_anomaly\_detection(X, y) method of loader\_nb.UrlDatasetLoader instance  
(X, y) --> X, y

This function perform unsupervised anomaly detection using Isolation Forest.

<https://practicaldatascience.co.uk/machine-learning/how-to-use-the-isolation-forest-model-for-outlier-detection>

```
help(loader.train_test_split)
```

Help on method train\_test\_split in module loader\_nb:

train\_test\_split(X, y, test\_size, random\_state, anomaly\_detection=True) method of loader\_nb.UrlDatasetLoader instance

This is a convenience method to train test split and have an option to perform anomaly detection or not after the split.

Read more in `sklearn.model_selection.train_test_split`

Parameters

-----

anomaly\_detection: True to perform unsupervised anomaly detection using Isolation Forest.

```
X_train, X_test, y_train, y_test = loader.train_test_split(X, y, test_size=TEST_SIZE, random_state=RANDOM_STATE)
```

The X\_train, y\_train shape:

```
(25694, 51)
```

```
(25694,)
```

The shape after unsupervised anomaly detection:

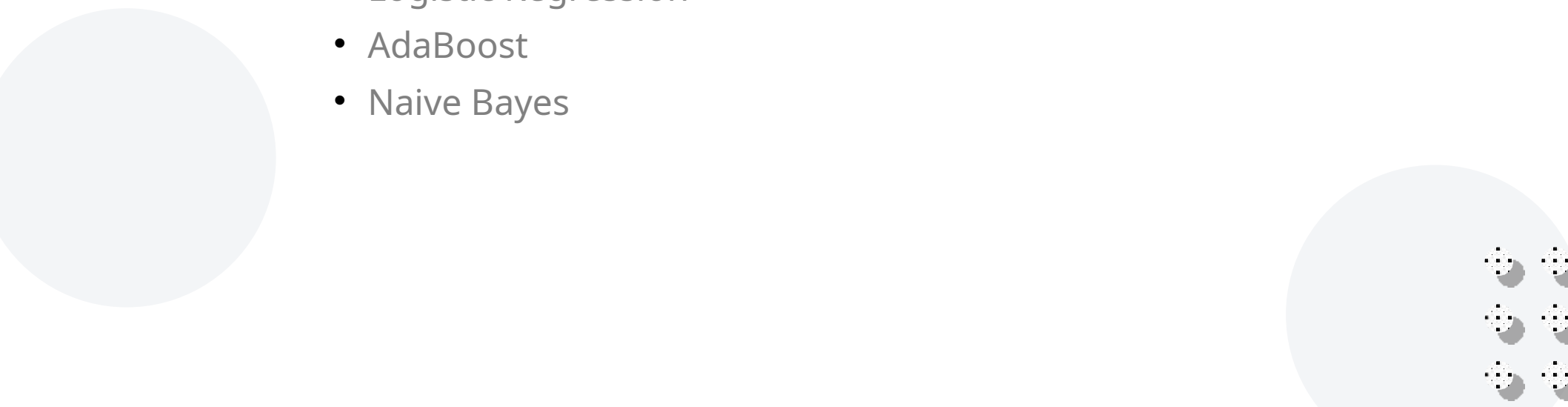
```
(25437, 51)
```

```
(25437,)
```

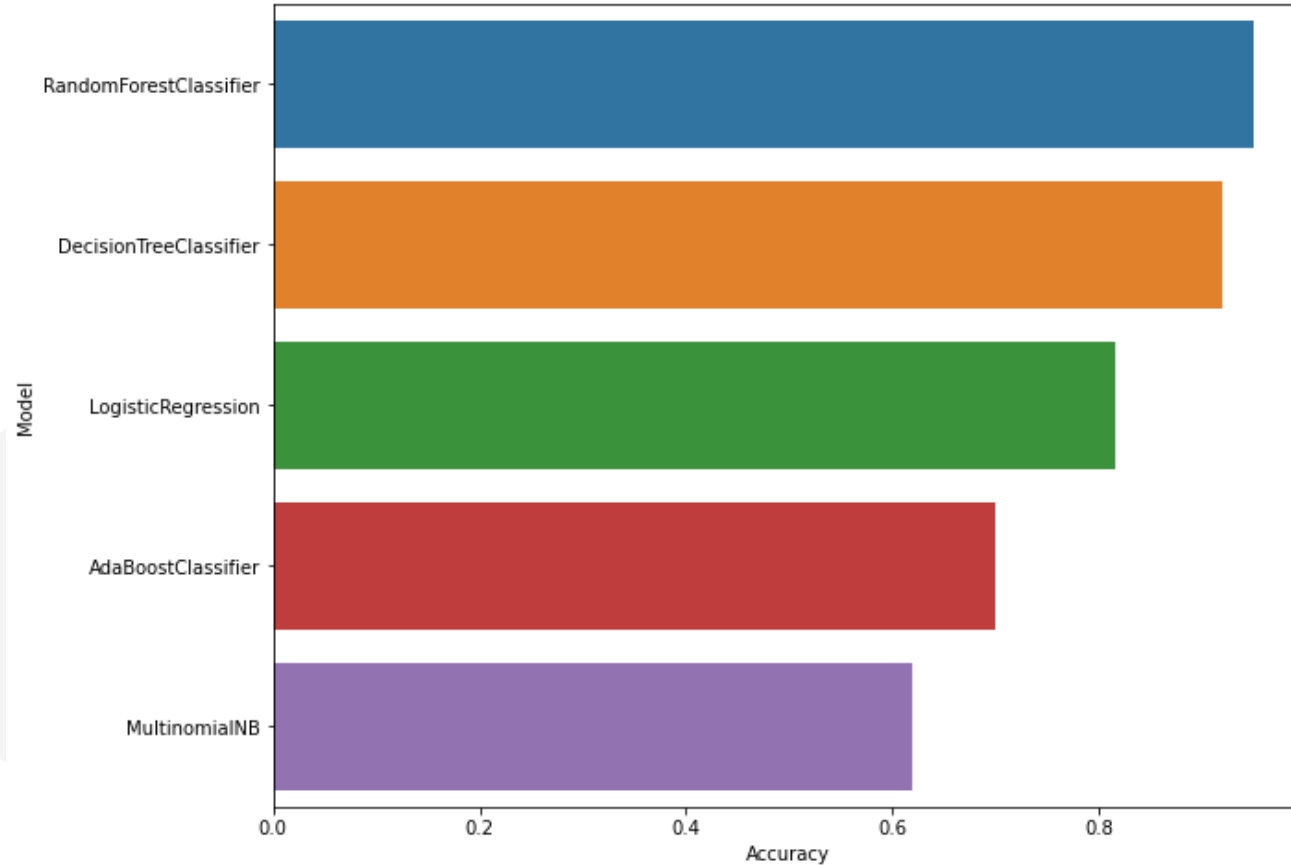
# Model Selection



**Using GridSearchCV with five folds we built five models**

- Random Forest
  - Decision Tree
  - Logistic Regression
  - AdaBoost
  - Naive Bayes
- 

# Results



	Model	Accuracy	F1-score
0	RandomForestClassifier	0.950376	0.950739
1	DecisionTreeClassifier	0.919464	0.920220
2	LogisticRegression	0.816639	0.815096
3	AdaBoostClassifier	0.698954	0.691719
4	MultinomialNB	0.618694	0.610399

## Conclusion

This study concludes that **Random Forest** is the best model to use to build a URL filter application using machine learning.

## Future Study

The focused of the study is the used of supervised learning algorithms. Future work could look on semi-supervised classification algorithms and new developing supervised algorithms.



Photo by [Dave Hoefler](#) on [Unsplash](#)

**Thank you**

Website:

<https://quickheaven.github.io/scs-3253-machine-learning/>

Github:

<https://github.com/quickheaven/scs-3253-machine-learning>

Team members

Arjie Cristobal

Omar Amjad

**Detection and categorization of  
Malicious URL's**

Url Prober Team  
05 April 2022



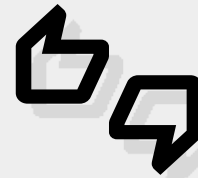
## 01 Lorem

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.



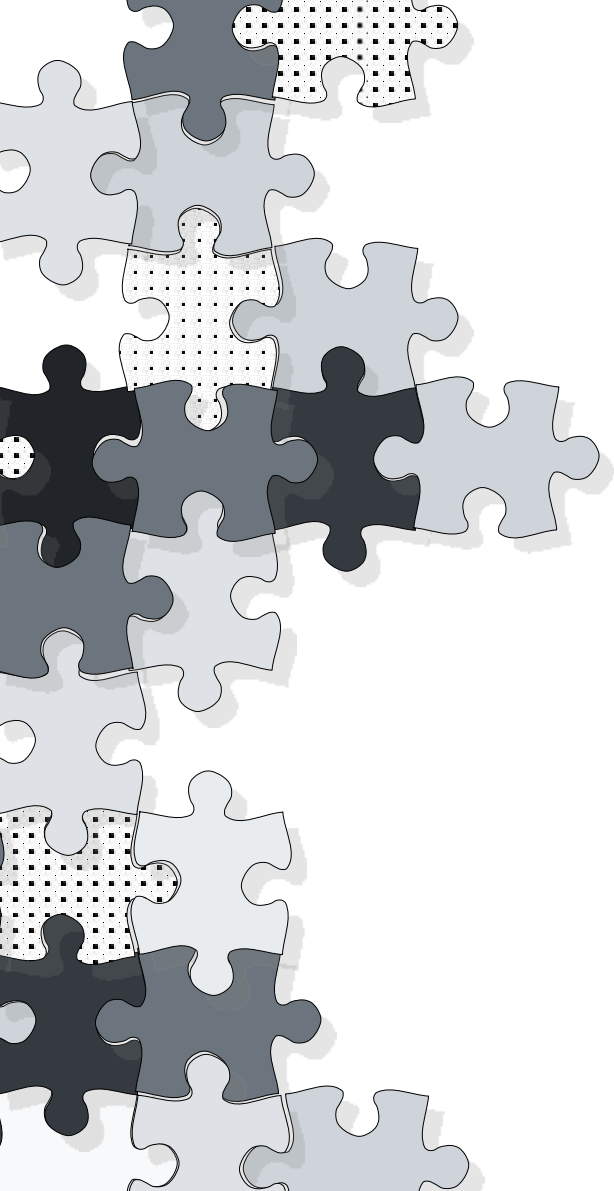
## 02 Ipsum

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.



## 03 Dolor

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.



“

Lorem ipsum dolor sit amet,  
consectetur adipiscing elit, sed do  
eiusmod tempor incididunt ut  
labore et dolore magna aliqua.

”

- Loremus

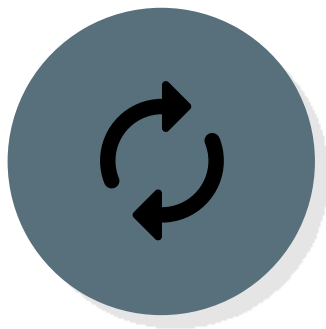


# Lorem & Ipsum



## Lorem

Lorem ipsum dolor sit amet,  
consectetur adipiscing elit,  
sed do eiusmod tempor  
incididunt ut labore et dolore  
magna aliqua.



## Ipsum

Lorem ipsum dolor sit amet,  
consectetur adipiscing elit,  
sed do eiusmod tempor  
incididunt ut labore et  
dolore magna aliqua.



## Dolor

Lorem ipsum dolor sit amet,  
consectetur adipiscing elit,  
sed do eiusmod tempor  
incididunt ut labore et dolore  
magna aliqua.

# 01

## Lorem

Lorem ipsum dolor sit amet,  
consectetur adipiscing elit, sed  
do eiusmod tempor incididunt ut  
labore et dolore magna aliqua.

# 03

## Dolor

Lorem ipsum dolor sit amet,  
consectetur adipiscing elit, sed  
do eiusmod tempor incididunt ut  
labore et dolore magna aliqua.

# 02

## Ipsum

Lorem ipsum dolor sit amet,  
consectetur adipiscing elit, sed  
do eiusmod tempor incididunt ut  
labore et dolore magna aliqua.

# 04

## Sit

Lorem ipsum dolor sit amet,  
consectetur adipiscing elit, sed  
do eiusmod tempor incididunt ut  
labore et dolore magna aliqua.