VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belgaum - 590 018

A COURSE PROJECT REPORT of
DATA MINING (18CSE54)

on

Spam URL Classification

Submitted in the Partial fulfillment of the requirements of Semester 5 of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING

Submitted by:

Mahesh B
1NT19CS105

Nikhil Rajput
1NT19CS127

Niraj Joshi
1NT19CS129

Womika Khan
1NT19CS218

Under the Guidance of

Ms. Vani Vasudevan
Prof., Dept. of CSE

NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY

Yelahanka, Bangalore - 560 064
Affiliated to Visveswaraya Technological University, Belgaum.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that Course Project Work titled "**Spam URL Classification**", carried out by **Mahesh B (USN: 1NT19CS105), Nikhil Rajput (USN: 1NT19CS127), Niraj Joshi (USN: 1NT19CS129) and Womika Khan (USN: 1NT19CS218)** bonafide students of **Nitte Meenakshi Institute of Technology** in partial fulfilment of Semester-5 of Bachelor of Technology Degree in Computer Science & Engineering under Visvesvaraya Technological University, Belagavi during the year 2021-2022. It is certified that all corrections/ suggestions indicated for Internal Assesment have been incorporated in the Report deposited in the Departmental Library. The Course Project Report has been approved as it satisfies the Academic requirements in respect of the Course Project Work prescribed for the said Degree.

Table of Contents

Abstract

One of the most exciting tools that have entered the material science toolbox in recent years is machine learning. This collection of statistical methods has already proved to be capable of considerably speeding up both fundamental and applied research. On the heels of the widespread adoption of web services such as social networks and URL shorteners, scams, phishing, and malware have become regular threats. Despite extensive research, email-based spam filtering techniques generally fall short for protecting other web services. To better address this need, we present Monarch, a real-time system that crawls URLs as they are submitted to web services and determines whether the URLs direct to spam. Social Networks such as Twitter, Facebook play a remarkable growth in recent years. The ratio of tweets or messages in the form of URLs increases day by day. As the number of URL increases, the probability of fabrication also gets increased using their HTML content as well as by the usage of tiny URLs. It is important to classify the URLs by means of some modern techniques. Conditional redirection method is used here by which the URLs get classified and also the target page that the user needs is achieved. Learning methods also introduced to differentiate the URLs and there by the fabrication is not possible. Also, the classifiers will efficiently detect the suspicious URLs using link analysis algorithm.

INTRODUCTION

1.1 Motivation

Malicious uniform resource locator (URL), i.e., Malicious websites are one of the most common cybersecurity threats. They host gratuitous content (spam, malware, inappropriate ads, spoofing, etc.) and tempt unwary users to become victims of scams (financial loss, private information disclosure, malware installation, extortion, fake shopping site, unexpected prize etc.) and cause loss of billions of rupees each year. The visit to these sites can be driven by email, advertisements, web search or links from other websites. In each case, the user must click on the malicious URL. The rising cases of phishing, spamming and malware has generated an urgent need for a reliable solution which can classify and identify the malicious URLs.

1.2 Problem Domain

Machine Learning (ML) which is subfield of data mining handles large scale well-formatted dataset efficiently. URL has been used and misused a lot to exploit the vulnerability of the user. This project focusses on classification of any URL as benign or malicious.

1.3 Aim and Objectives

The main idea behind the proposed system after reviewing the above papers was to create a Spam URL Classification model based on the data. We analyzed the classification algorithms namely Decision Tree and Logistic Regression. We use ROC-AUC score to check our accuracy(TPR and FPR)

DATA SOURCE AND DATA QUALITY

2.1 Dataset Used

The dataset used was the Spam URL Classification, which is a combination of 4 different database, but only the UCI Cleveland dataset was used. This database consists of a total of 2 attributes that is URL and target if it is true or false. Therefore, we have used the already processed Spam URL Classification dataset available in the Kaggle website for our analysis.

2.2 Data Preprocessing

- Normalization : Normalization is the process of reorganizing data in a database so that it meets two basic requirements: There is no redundancy of data, all data is stored in only one place.
- Binarization : Binarization is the process of transforming data features of any entity into vectors of binary numbers to make classifier algorithms more efficient.

METHODS AND MODELS

3.1 Data Mining Questions

Based on the given data set we will be able to classify if the given URL is spam or not, based on the criterion like

- URLs length
- URLs Digit count
- URLs Number of Words

3.2 Data Mining Algorithms

- **Logistic Model Tree Induction (LMT):** In computer science, a logistic model tree (LMT) is a classification model with an associated supervised training algorithm that combines logistic regression (LR) and decision tree learning.

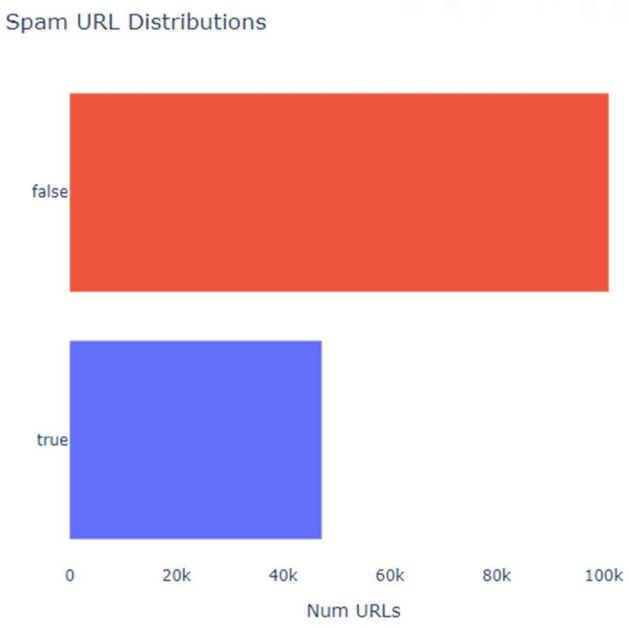
3.3 Data Mining Models

- **Logistic Regression**: Linear regression is a linear ML algorithm which is used for classification. In logistic regression, the probabilities of possible classes are calculated using the sigmoid function. The sigmoid function is used because the range of the function is from 0 to 1. Logistic regression is used to understand the relationship between independent and dependent variables. It is easy to implement and most computationally efficient algorithm among those compared in this paper. Logistic Regression can be used in the case of binomial classification. It assumes
that the independent variables are uncorrelated.

MODEL EVALUATION AND DISCUSSION

In [2]:

```
vc = df['is_spam'].value_counts().to_frame().reset_index().head(15)
fig = px.bar(x=vc["is_spam"][::-1], y=vc["index"][::-1], orientation='h', color=vc['index'])
fig.update_layout(title = "Spam URL Distributions", xaxis_title="Num URLs", yaxis_title = "", wi
dth=600, plot_bgcolor="#fff", showlegend = False)
fig.show()
```



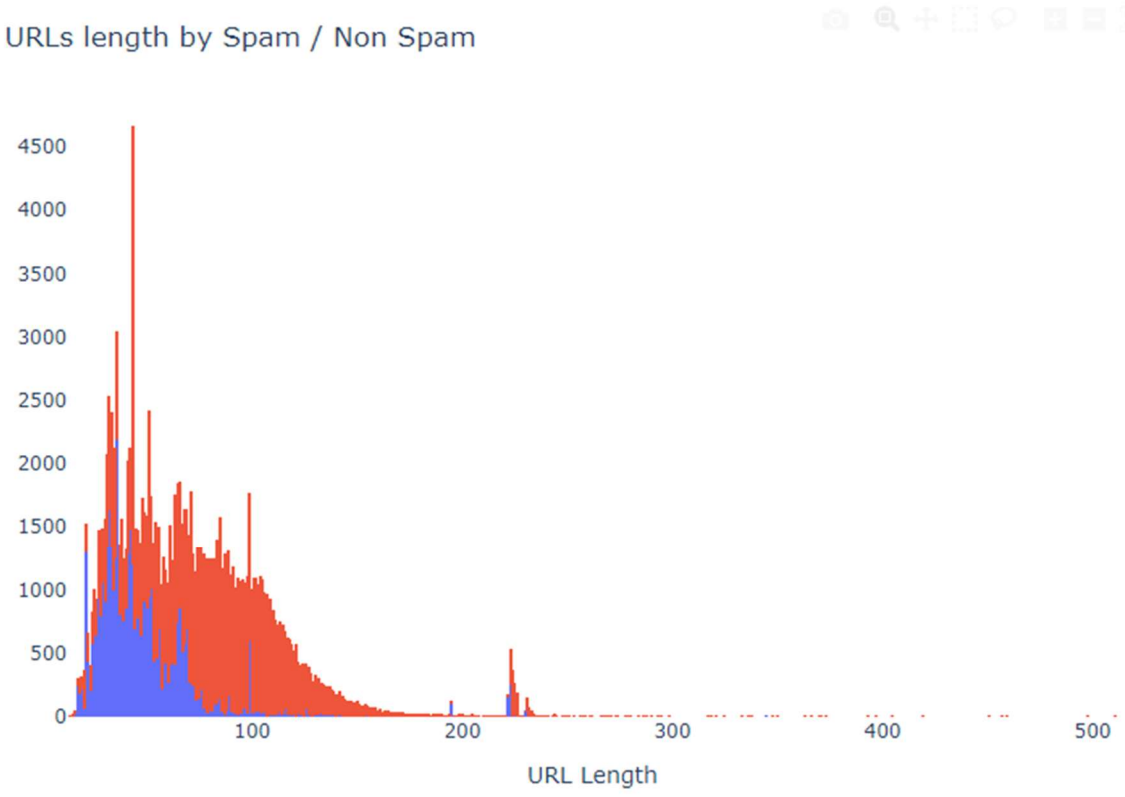
The following is a classification of the URLs without considering any factors like number of digits or special characters in the URL

Let's create a few features

In [3]:

```
df['len_url'] = df['url'].apply(lambda x : len(x))
df['contains_subscribe'] = df['url'].apply(lambda x : 1 if "subscribe"
in x else 0)
df['contains_hash'] = df['url'].apply(lambda x : 1 if "#" in x else 0)
df['num_digits'] = df['url'].apply(lambda x : len("".join(_ for _ in x
if _.isdigit())) )
df['non_https'] = df['url'].apply(lambda x : 1 if "https" in x else 0)
df['num_words'] = df['url'].apply(lambda x : len(x.split("/")))
df.head()
```

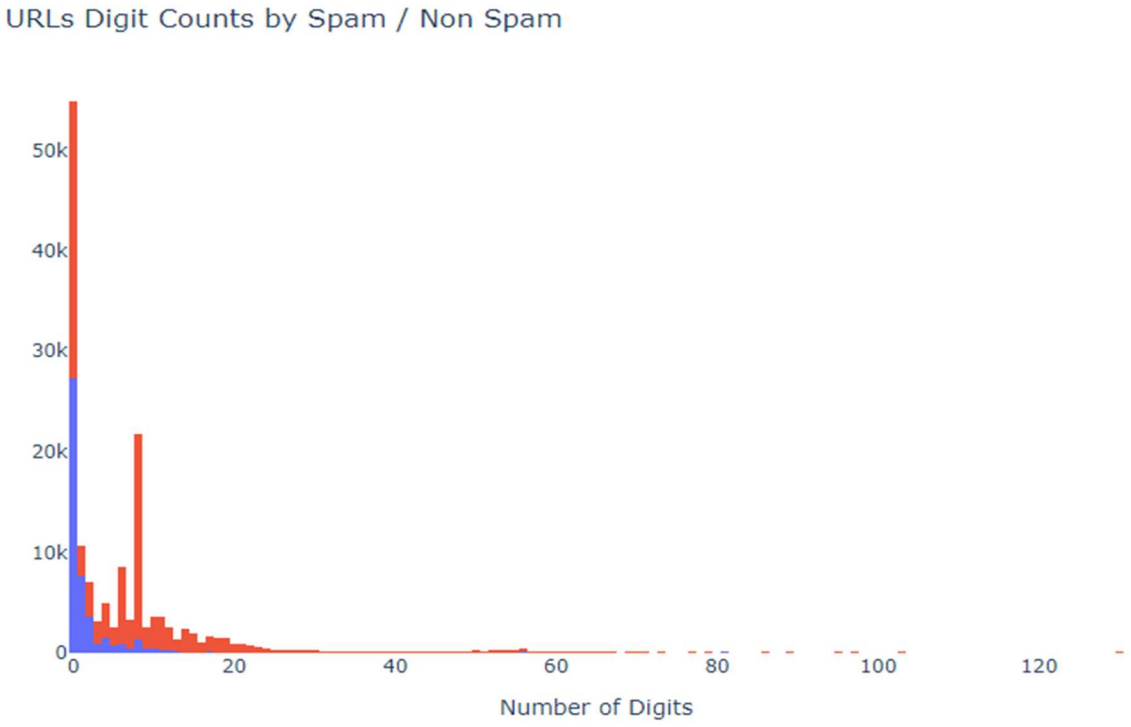
Out[3]:



The above Bar Graph represents if the URL is spam or not based on URL length.

In [5]:

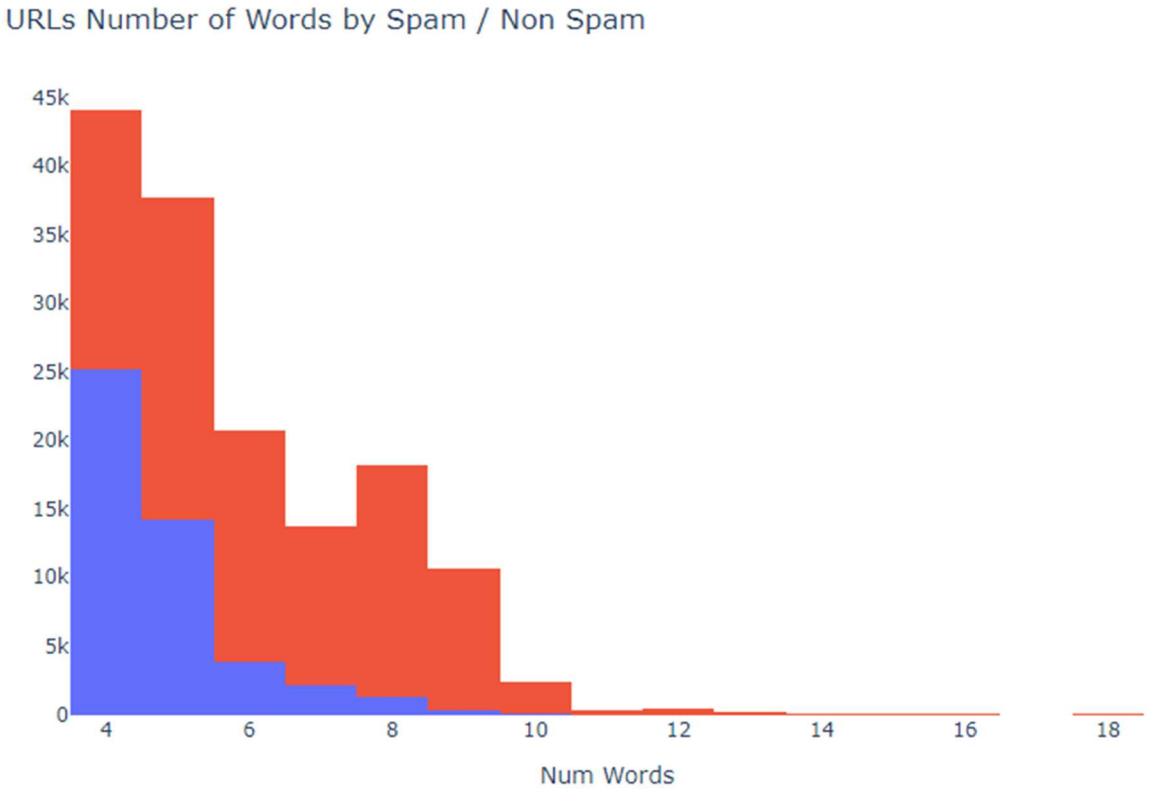
```
fig = px.histogram(df, x="num_digits", color="is_spam")
fig.update_layout(title = "URLs Digit Counts by Spam / Non Spam", xaxis
_title="Number of Digits", yaxis_title = "", plot_bgcolor="#fff", showl
egend = False)
fig.show()
```



The above Bar Graph represents if the URL is spam or not based on Number of Digits.

In [6]:

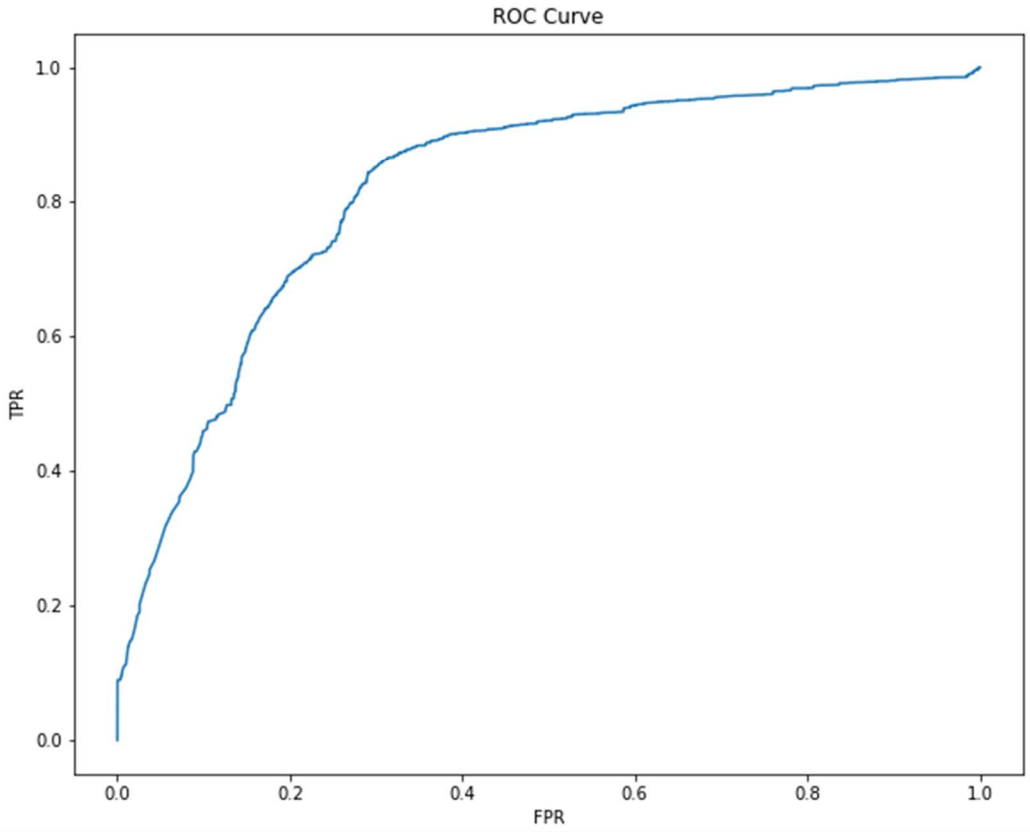
```
fig = px.histogram(df, x="num_words", color="is_spam")
fig.update_layout(title = "URLs Number of Words by Spam / Non Spam", xa
xis_title="Num Words", yaxis_title = "", plot_bgcolor="#fff", showlegen
d = False)
fig.show()
```



```
plt.figure(figsize=(10,8))
fpr, tpr, tresholds = roc_curve(y_test, test_probas)
plt.plot(fpr, tpr)
plt.title('ROC Curve')
plt.xlabel('FPR')
plt.ylabel('TPR')

print('ROC-AUC-score: ', roc_auc_score(y_test, test_probas))
```

```
ROC-AUC-score:  0.8209149813245185
```



We use ROC-AUC score to check our accuracy (TPR and FPR).

The result of this analysis indicates that the logistic regression algorithm is the most efficient algorithm with an ROC AUC accuracy score of 82.10% for prediction of spam URL.

CONCLUSION AND FUTURE DIRECTION

Recently, with the increase in Internet usage, cybersecurity has been a significant challenge for computer systems. Different malicious URLs emit different malicious software and try to capture user information. Signature-based approaches have often been used to detect such websites and detected malicious URLs have been attempted to restrict access by using various security components. We get an AOC RUC score of 0.821 which is plotted against TPR and FPR. The results show that being able to identify spam websites based on URL alone and classify them as spam URLs without relying on page content will result in significant resource savings as well as safe browsing experience for the user.
Machine learning techniques were used in this work to process raw data and provide a new and novel discernment towards spam URL prediction. Spam URL prediction is challenging and very important in this age of cybersecurity

In future the work can be enhanced by developing a web application based on Logistic Regression as well as using a larger dataset as compared to the one used in this analysis which will help to provide better results to ensure safe browsing of the users.

6. Reflection Portfolio

Data mining has a direct impact on the safe browsing of the users. The aspects of data mining are the ones that ensures safe searching of the users. Through this learning activity we have learned how to apply different data mining algorithm models to generate graphical representations and meaningful patterns.

REFERENCES

- **Kaggle**
- **Malicious URL Detection A Comparative Study**

APPENDICES

a. Link to the dataset chosen

Spam URL Classification Dataset

b. Python Codes Implemented

The Jupyter notebook is attached with the word document.

```
import plotly.express as px
import pandas as pd
df = pd.read_csv("../input/spam-url-prediction/url_spam_classification.csv")
df.head()
```

```
vc = df['is_spam'].value_counts().to_frame().reset_index().head(15)
fig = px.bar(x=vc["is_spam"][::-1], y=vc["index"][::-1], orientation='h', color=vc['index'])
fig.update_layout(title = "Spam URL Distributions", xaxis_title="Num URLs", yaxis_title = "",
width=600, plot_bgcolor="#fff", showlegend = False)
fig.show()
```

```
df['len_url'] = df['url'].apply(lambda x : len(x))
df['contains_subscribe'] = df['url'].apply(lambda x : 1 if "subscribe" in x else 0)
df['contains_hash'] = df['url'].apply(lambda x : 1 if "#" in x else 0)
df['num_digits'] = df['url'].apply(lambda x : len("".join(_ for _ in x if _.isdigit())) )
df['non_https'] = df['url'].apply(lambda x : 1 if "https" in x else 0)
df['num_words'] = df['url'].apply(lambda x : len(x.split("/")))
df.head()
```

```
fig = px.histogram(df, x="len_url", color="is_spam")
fig.update_layout(title = "URLs length by Spam / Non Spam", xaxis_title="URL Length", yaxis_title = "",
plot_bgcolor="#fff", showlegend = False)
fig.show()
```

```
fig = px.histogram(df, x="num_digits", color="is_spam")
fig.update_layout(title = "URLs Digit Counts by Spam / Non Spam", xaxis_title="Number of Digits",
yaxis_title = "", plot_bgcolor="#fff", showlegend = False)
fig.show()
```

```
fig = px.histogram(df, x="num_words", color="is_spam")
fig.update_layout(title = "URLs Number of Words by Spam / Non Spam", xaxis_title="Num Words",
yaxis_title = "", plot_bgcolor="#fff", showlegend = False)
fig.show()
```

```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from sklearn.pipeline import Pipeline
import matplotlib.pyplot as plt
from sklearn.metrics import roc_auc_score, roc_curve

target = 'is_spam'
features = [f for f in df.columns if f not in ["url", target]]
X_train, X_test, y_train, y_test = train_test_split(df[features], df[target], test_size=0.2,
random_state=0)

sc = MinMaxScaler()
clf = LogisticRegression()
pipe_lr = Pipeline([('scaler', sc), ('clf', clf)])
pipe_lr.fit(X_train, y_train)

test_probas = pipe_lr.predict_proba(X_test)[:,1]

plt.figure(figsize=(10,8))
fpr, tpr, tresholds = roc_curve(y_test, test_probas)
plt.plot(fpr, tpr)
plt.title('ROC Curve')
plt.xlabel('FPR')
plt.ylabel('TPR')

print('ROC-AUC-score: ', roc_auc_score(y_test, test_probas))
```

```
ROC-AUC-score:  0.8209149813245185
```