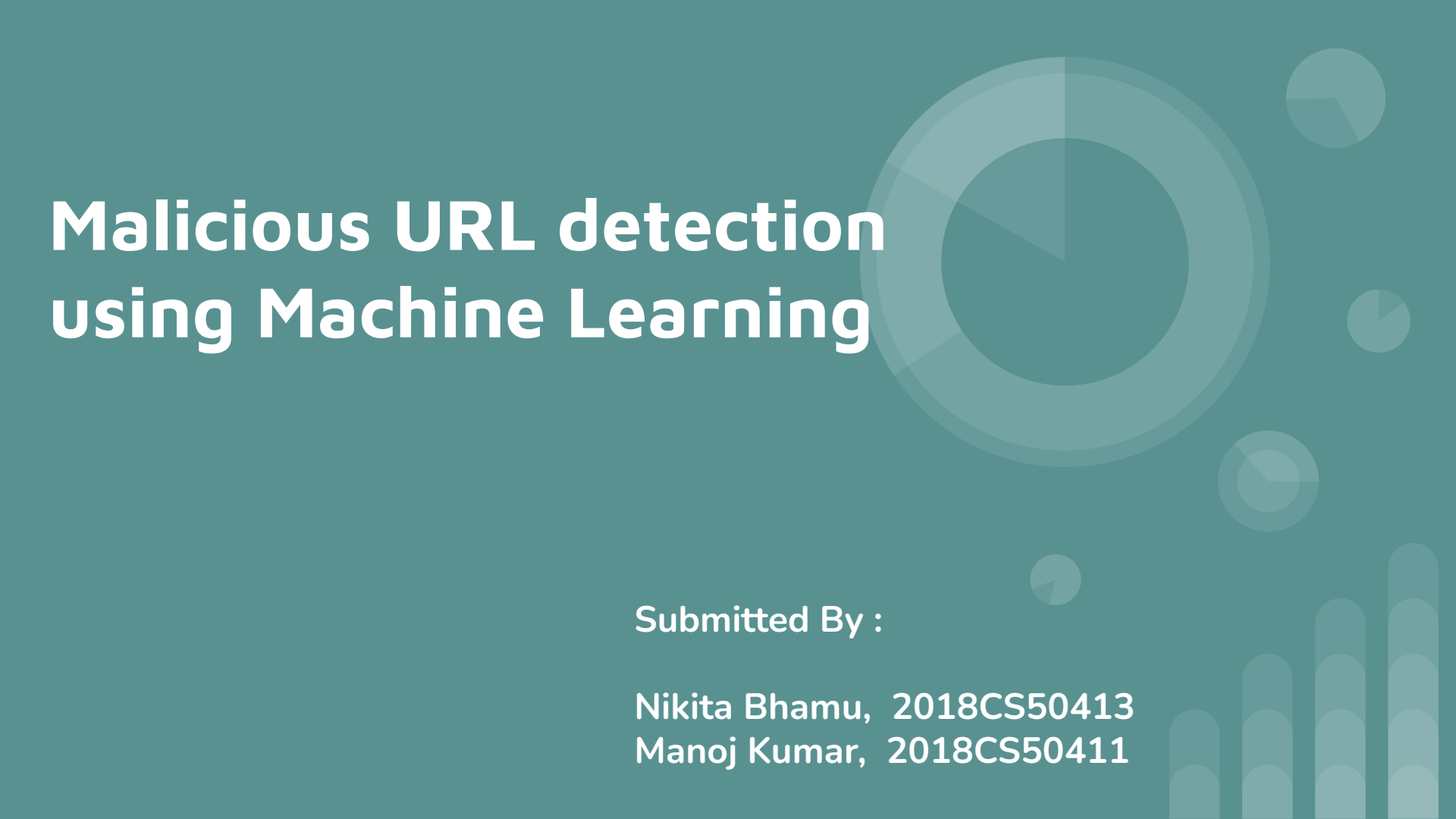


Malicious URL detection using Machine Learning



Submitted By :

Nikita Bhamu, 2018CS50413
Manoj Kumar, 2018CS50411



Introduction

- Now a days, internet is a need of every person and every business.
- Technology advancement made many services very easy like:
 - Online Banking
 - E-Commerce
 - Social Networking
- Technology advancement also brought frauds:
 - Financial Frauds (bank details compromise)
 - Non-financial frauds (Identity Theft)
- Using malicious URLs, they can execute code on users' computer to:
 - Malware download
 - Redirect to unwanted or other phishing sites



Background and Related Work

- **Signature based malicious URL Detection / Blacklist method:**
 - A large signature set of malicious URLs is stored in a database.
 - A database query is executed to find that URL from the set of malicious URLs.
 - If that URL exists in the database, that means that the URL is malicious.
 - Else that URL is safe and user can proceed with that URL.
- **Machine Learning Based Malicious URL Detection:**
 - Uses a set of malicious URLs as training data.
 - Learns a prediction function to classify a URL as malicious or safe based on the statistical properties of the data.
 - It removes the disadvantage of the blacklisting method as it also predicts for new URLs.
 - Some classification algorithms are Naive Bayes, SVM, logistic regression etc.



Background and Related Work

- **Malicious URL Detection Tools:**
 - **URL Void:** uses multiple engines and blacklists of domains.
 - Advantage: Compatibility (supports many testing services and browsers).
 - Disadvantage: Heavily depends on the given set of signatures.
 - **UnMask Parasites:** It do the following for detecting the malicious URL:
 - Downloads the data of provided links
 - Parses the HTML codes, javaScripts, external links and iframes and analyze if something strange is happening on the site.
 - **Dr. Web Anti-Virus Link Checker:** an add-on for browsers like Chrome, Firefox, Opera and Microsoft Edge to automatically find and scan malicious content on download links on all social networking links such as Facebook, Google+ etc.
 - **Some Other Tools:** There are some more URL checking tools such as UnShorten.it, VirusTotal, Norton Safe web, McAfee SiteAdvisor, Google Safe Browsing etc.



Problem Statement

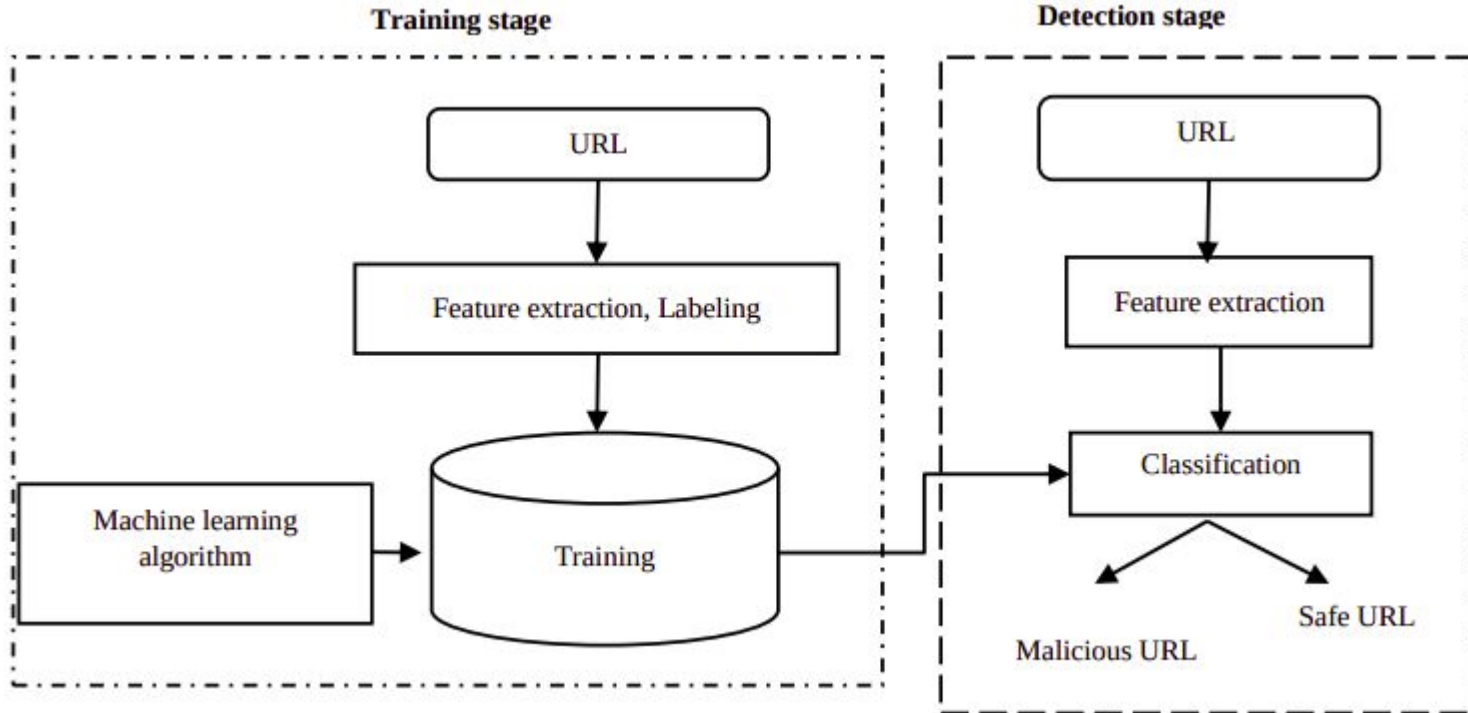
The problem can be formulated as a binary classification problem that classifies the URL either as malicious(-1) or as benign(+1) on the basis of a labelled URLs dataset.

The problem has the following two steps :-

1. Feature extraction :
All the given URLs need to be represented as the feature vectors which takes into account features playing a role in the URL being malicious or benign.
2. Classification using machine learning algorithms :
We need to use those feature vectors obtained in the first step to train the classifier and finally produce the classification result.

We planned to analyse the importance of various features as well as the efficiency of various machine learning models in classifying the URLs correctly.

Solution Plan





Dataset Description :-

The dataset we are using is taken from Kaggle

<https://www.kaggle.com/datasets/antonyj453/urldataset>

The dataset has the following two attributes:

1. URL
2. Label : Benign or malicious

There are a total of 4,20,000 entries(rows) in the table.

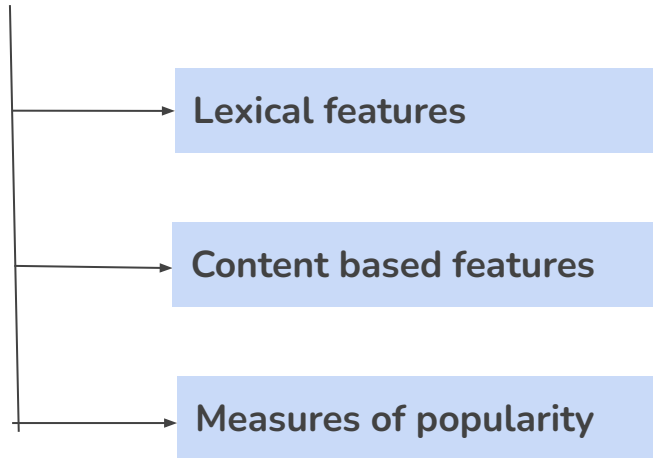
We randomly choose 80% of this dataset to train our models and the remaining 20% is used for the testing purpose.

Step 1 : Extraction of features



SOLUTION PROPOSED EARLIER :-

We planned to use the following three types of features of a URL to train the machine learning models



SOLUTION IMPLEMENTED :-



The content based features and the feature based on measures of popularity both needs to download the complete html page of the URL.

Since we have millions of training data, and extracting the both these features of all training URLs is neither memory feasible nor speed efficient,
So to make our solution more realistic to apply, we moved ahead with Lexical features.

Lexical features



Content based features



Measures of popularity



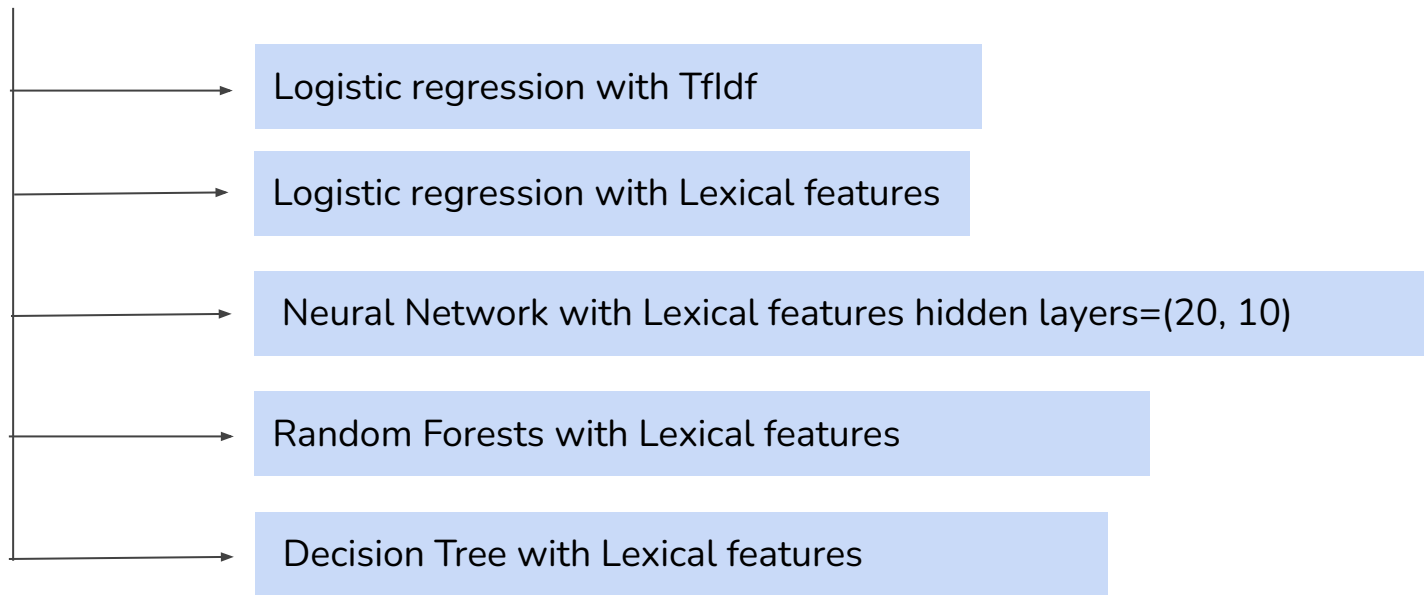
The following Lexical Features are considered :-



1. Number of character '.' in URL
2. Number of subdomain levels
3. The depth of URL
4. The length of URL
5. Number of the dash character '-'
6. Number of dash character in the hostname
7. There exists a character '@' in URL
8. There exists a character '~(tilde) in URL
9. Number of the underscore character
10. Number of the character '%'
11. Number of the character '&'
12. Number of the character '#'
13. Number of the numeric characters
14. Check if the IP address is used in the hostname of the website URL
15. Length of hostname
16. Length of the link path
17. Number of sensitive words (i.e., "secure", "account", "webscr", "login", "ebayisapi", "sign in", "banking", "confirm") in website

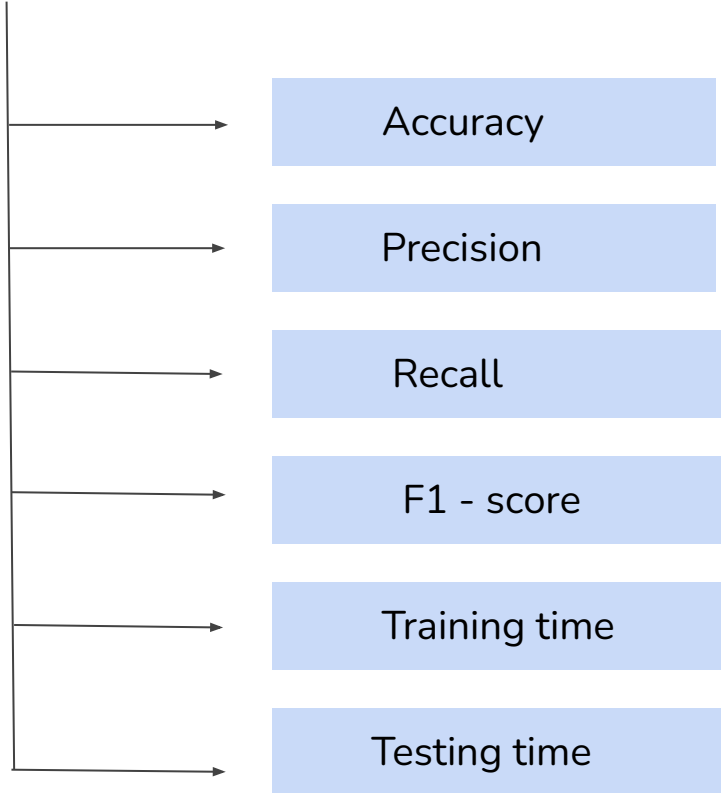
Step 2 : Classification using machine learning models

We have used the following machine learning models for the classification :

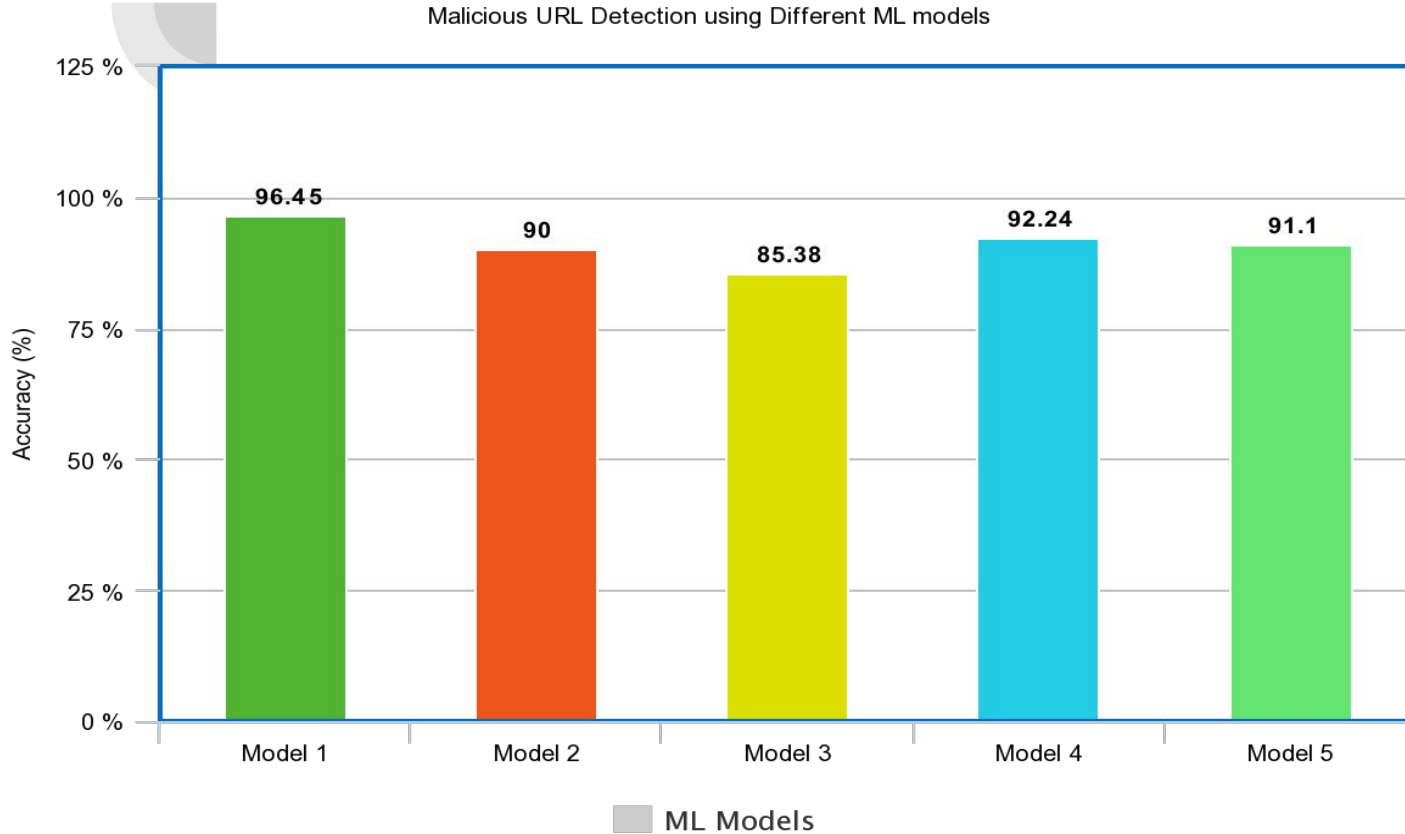


Evaluations :

The models are evaluated on the basis of the following parameters



ACCURACY :



1. Logistic Regression with Tfldf

2. Neural Network with Lexical Features (hidden-layer size=(20,10))

3. Logistic Regression on Lexical Features

4. Random Forests on Lexical Features

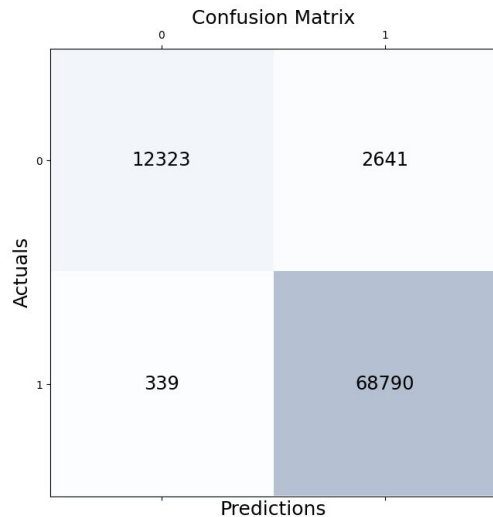
5. Decision Tree with Lexical Features



Confusion Matrix

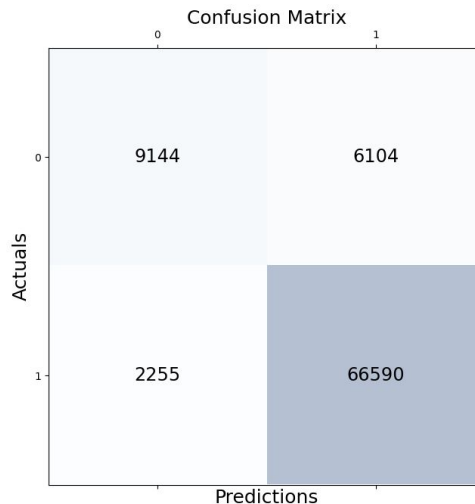
1. Logistic Regression with TfIdf

training time: 14.728665590286255
testing time: 0.005624294281005859



2. Neural Network with Lexical Features (hidden-layer size=(20,10))

training time: 122.2729115486145
testing time: 0.07317352294921875

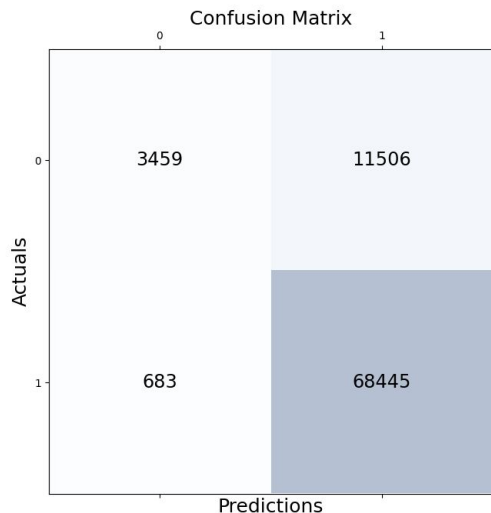




Confusion Matrix

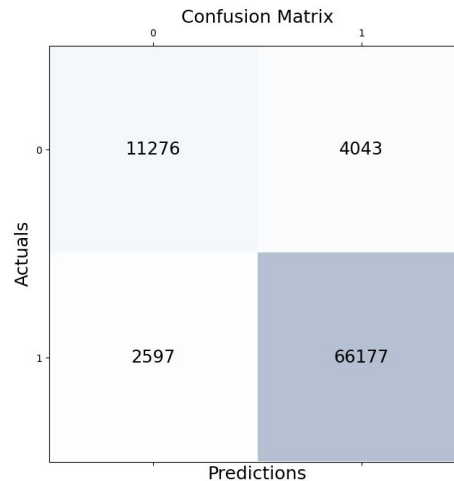
3. Logistic Regression on Lexical Features

training time: 0.881702184677124
testing time:
0.0025887489318847656



4. Random Forests on Lexical Features

training time: 138.43354725837708
testing time: 1.9103291034698486





Confusion Matrix

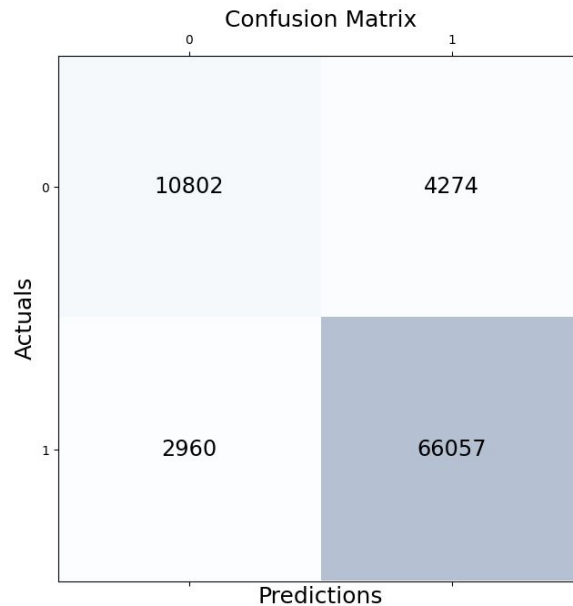
5. Decision Tree with Lexical Features

training time:

3.0172131061553955

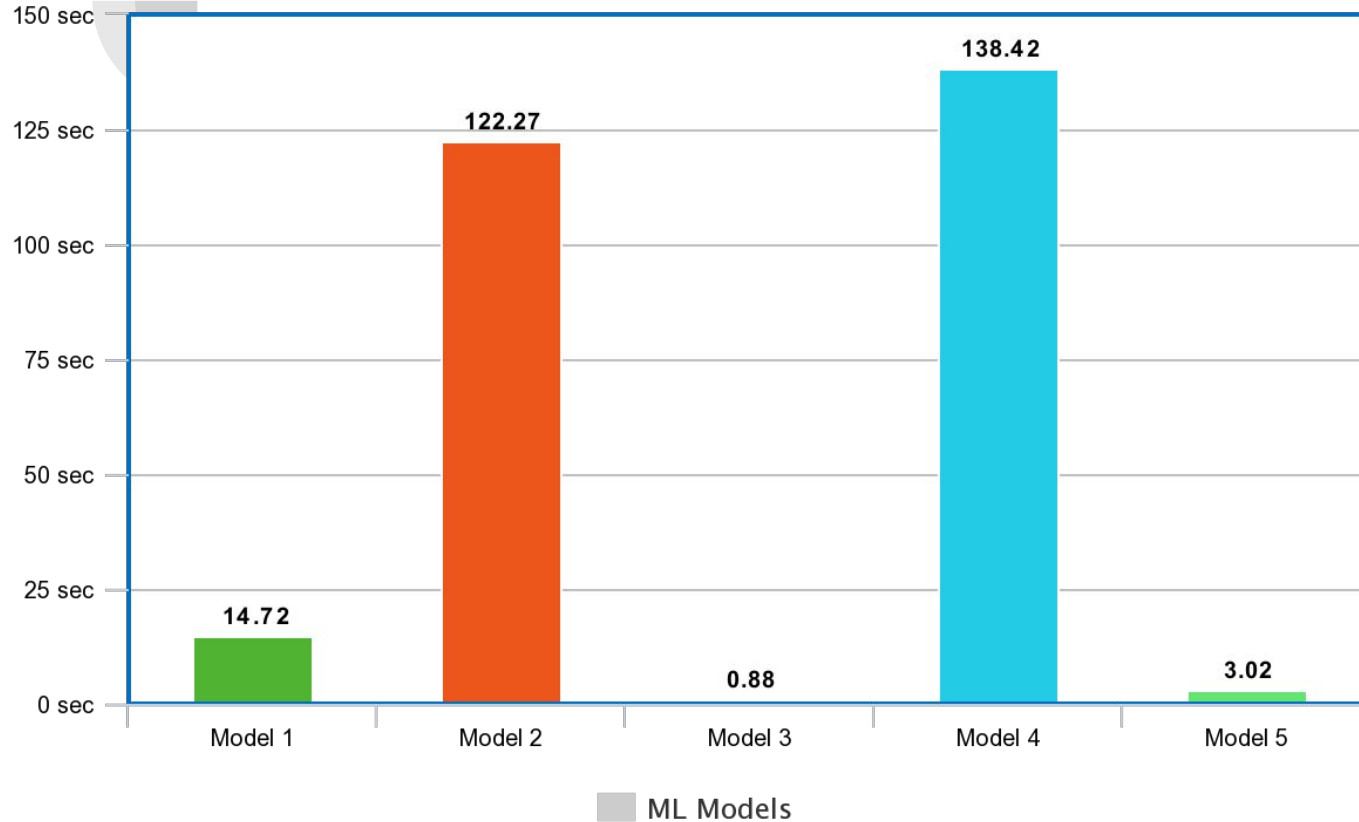
testing time:

0.22769427299499512



Training Time :

Training Time of ML Models



1. Logistic Regression with TfIdf

2. Neural Network with Lexical Features (hidden-layer size=(20,10))

3. Logistic Regression on Lexical Features

4. Random Forests on Lexical Features

5. Decision Tree with Lexical Features

Model	Accuracy	Precision	Recall	F1-score	Training time (sec)	Testing time (sec)
Logistic Regression with TfIdf	96.45	0.96	0.99	0.98	14.72	0.005
Neural Network with lexical features (hidden-layer size=(20,10))	90	0.92	0.96	0.94	122.27	0.073
Logistic Regression with lexical features	85.38	0.85	0.99	0.92	0.88	0.002
Random forests with lexical features	92.24	0.94	0.96	0.95	138.43	1.91
Decision tree with lexical features	91.10	0.94	0.95	0.95	3.017	0.227



Conclusion :

- All the models used for classification have accuracy more than 85%.
- The best performing model in the URL classification is Logistic Regression with Tf-Idf data having an accuracy of 96.45%.
- Model of Logistic regression with tokenization and Tf-Idf works best on the given dataset. It is because it calculates the weight of the words based on the inbuilt python dictionary and there are certain fixed words in the benign url such as https etc. which is helping in this model to make the predictions correct.
- The least time in training and testing is also taken by the Logistic Regression with Lexical Features.



- Out of the 5 selected models the worst performing model is Logistic Regression with lexical features.
- Logistic Regression with lexical features performs bad than the other models as Regression is a simple model and if the features passed in the regression model increases, its efficiency decreases and since we have considered 17 lexical features which is indeed very large for a regression model to work accurately.
- Machine Learning approaches are important in detection of Malicious URLs as they can predict newer URLs as well.

Code is available at :

<https://github.com/manoj2601/Malicious-URL-Detection-using-Machine-Learnin>

g



Thank You!