A Project Report

on

Malicious URL Detection Using Machine Learning

Aryan Nandu(A20583853)
Rudra Patel (A20594446)

Under the guidance of

Prof. Oleksandr Narykov

Department of Computer Science
Illinois Institute Of Technology
Chicago, Illinois – 60616

Acknowledgement

I would like to express my sincere gratitude to all those who have supported and guided me throughout the process of conducting this report on Malicious URL Detection Using Machine Learning . This endeavor would not have been possible without their valuable contributions and assistance.

We are thankful to our college Illinois Institute of Technology for considering our project and extending help at all stages needed during our work of collecting information regarding the project.

We take this opportunity to express our profound gratitude and deep regards to our guide Prof. Oleksandr Narykov for her exemplary guidance, monitoring and constant encouragement throughout the course of this project.

This work would not have been possible without the collective efforts of these individuals and organizations. While any shortcomings in this report are solely our responsibility, their contributions have significantly enriched its content.

Abstract

The detection of malicious URLs is a critical challenge in ensuring cybersecurity, given the increasing sophistication of cyber threats. This report presents the development and evaluation of a predictive model for malicious URL detection using Machine Learning, where a dataset is categorized as benign and malicious. The study involved extensive exploratory data analysis and visualization to uncover patterns and insights, aiding in feature selection and preprocessing.

Five machine learning classifiers—Logistic Regression, Decision Trees, Random Forest, GaussianNB, and ExtraTrees—were implemented and evaluated on the dataset. The models were assessed based on their performance metrics: accuracy, precision, recall, and F1-score. Visualization techniques, such as heatmaps and feature distributions, provided a deeper understanding of the data and the models' behavior. The comparative analysis highlighted the strengths and weaknesses of each classifier, enabling the identification of the most effective approach for malicious URL detection.

The findings demonstrate the potential of machine learning algorithms to distinguish between benign and malicious URLs with high accuracy. A dataset of 651,191 is taken from Kaggle where 428,103 are Benign and 223,088 are Malicious. This work contributes to cybersecurity by showcasing a systematic approach to model development and evaluation, offering a scalable solution for real-world threat detection.

Keywords: [Cybersecurity, Malicious, Benign, Machine Learning]

Table of Contents

List of Figures

Chapter 1

Introduction

Android is the world's most popular operating system, with more than 2.5 billion active users and an estimated 75% share of the mobile device market. Among the threats faced by Android users, malicious URLs remain a significant concern, as they can lead to phishing attacks, malware installations, and data theft. Detecting such malicious URLs effectively is crucial in safeguarding users against these risks.

In this project, a robust prediction model was developed to identify malicious URLs using a comprehensive dataset of 651,191 URLs labeled as benign or malicious. The study addresses the limitations of traditional methods, such as blacklists and static feature classification, which often fail to meet the desired standards of accuracy and adaptability. By leveraging machine learning techniques, the proposed system extracts lexical and behavioral features from URLs and uses them to train predictive models.

Five classifiers—Logistic Regression, Decision Tree, Random Forest, Support Vector Machine (SVM), and Gradient Boosting—were implemented to classify URLs as malicious or benign. The performance of these classifiers was rigorously evaluated, and the most accurate and computationally efficient model was identified for real-world deployment. This project emphasizes the importance of machine learning in enhancing URL threat detection, providing a scalable and precise solution to combat cyber threats.

Github Link - DPA Project

Github Link - https://github.com/Aryan2210/DPA$project$

Chapter 2

Literature Review

Sanchit Goel [1] This paper addresses the growing problem of QR code-related fraud in emerging markets. It proposes a solution called "Secure QR (sQR)" using deep learning models and machine learning models. After extensive testing, the Vision Transformer model was found to be effective in detecting malicious QR codes, achieving more than 99.96 and sensitivity. Lightweight now has the ability to integrate with existing QR systems, especially in areas where there is a lot of crime, such as India's UPI QR code. Regular training with updated information ensures that the system remains resilient to changing threats.

Saleem Raja [2] This research addresses the major threat from cybercriminals using malicious URLs on Internet services. Heuristic research using features extracted from data and machine learning algorithms, including support vector classifiers, naive Bayes, logistic regression, K-nearest neighbors, and random forest. The results demonstrate the effectiveness of the simple method and highlight the superiority of the k-NN classifier in terms of accuracy and efficiency. This new approach shows promise in combating cyber threats in the online space.

Shantanu [3] This article discusses the increase in cyber threats, especially among online businesses, during the Covid-19 pandemic. It uses Kaggle datasets to test various machine-learning methods to detect malicious URLs. Random Forest is the best with an F1 score and an accuracy rate of almost 99.6 %. To improve its performance, future efforts could focus on training the classifier on more balanced data that balances between malicious and malicious websites.

Sara Afzal [4] This research uses URL shortening programs to combat online threats such as phishing attacks and solve privacy issues in online social networks (OSNs). This method combines a semantic vector model, URL encryption, and backward neural network (LSTM) with k-means clustering for classification. The results showed good accuracy; It achieved

98.3for token-level injection and URL encryption, laying a solid foundation for on-the-fly URL detection and possible future improvements using deep learning models.

Cho Do Xuan [5] This article addresses the issue of faulty URLs in the digital space and highlights the need for improved search engines. This study adopted a signature-based and machine learning-based approach using Support Vector Machine (SVM) and Random Forest (RF) algorithms. The 100-tree RF model showed good results, achieving high accuracy and low error rates. While acknowledging inconsistent data and recommending future work, this paper could benefit from additional reasoning behind algorithm selection. Overall, this study provides the best solutions and user tools to improve network security against malicious URLs.

2.1 Problem Statement and Objective

Problem Statement

As digitalization continues to grow, the use of links or URLs has become integral to accessing online services, information, and communication. However, the rise in URL usage has also led to a surge in cyber threats, particularly from malicious URLs that deceive users into visiting harmful websites. These malicious links are often disguised as legitimate ones and are used for phishing, malware distribution, and various other forms of cyberattacks, putting user data and security at risk. To combat this, the goal is to develop a malicious URL detection system that can analyze URLs and predict whether they are benign or harmful. By utilizing machine learning techniques, the system will be trained on a large dataset of both malicious and benign URLs to identify patterns and features indicative of threats. This automated detection system will provide an efficient, real-time solution to help users avoid potential cyber dangers and enhance online safety.

Objective

- Develop a Machine Learning Model: Create a machine learning-based model capable of classifying URLs as either benign or malicious based on their features and patterns.
- Feature Extraction and Analysis: Identify and extract relevant features from URLs, such as domain, path, query parameters, and other lexical aspects, to effectively distinguish between malicious and benign links.
- Train and Evaluate Multiple Classifiers: Implement and compare various machine learning classifiers to identify the most accurate and efficient model for malicious URL detection.
- Improve Adaptability: Ensure the detection system can continuously learn and adapt to new and emerging malicious URL patterns, maintaining its relevance in combating

evolving cyber threats.

- Visualization of Results: Use various visualization techniques to display the performance of the model, highlight key features, and showcase the detection system's accuracy and effectiveness.

2.2 Scope

The primary objective of this project is to develop a Malicious URL Detection System using machine learning techniques to address the growing threat of malicious URLs. The system will classify URLs as either benign or malicious based on a variety of features extracted from the URLs themselves. A comprehensive dataset containing both malicious and benign URLs will be collected to train and optimize the machine learning model, with the goal of achieving a detection accuracy of 90% or higher.

The machine learning model will be rigorously evaluated using various performance metrics to ensure its effectiveness in detecting malicious URLs. The model will be continually updated to adapt to emerging threats, ensuring that the detection system stays current with new and evolving attack techniques.

Additionally, the project will focus on user education regarding safe URL browsing practices, providing essential guidelines to help users recognize and avoid malicious URLs. The system will prioritize user privacy and ensure that no personal data is collected or compromised during the analysis process.

The ultimate aim of the project is to offer a reliable, accurate solution that protects users from cyber threats such as data theft, financial loss, and other malicious activities associated with harmful URLs.

Chapter 3

Proposed System

3.1 Analysis/Framework/ Algorithm

Machine Learning Classifiers:

- **Decision tree**: - A Decision Tree is a machine-learning model that uses a tree-like structure to make decisions or predictions. It splits the data into subsets based on the most important attributes at each level, leading to a final decision at the leaf nodes. It's a transparent and interpretable algorithm used for classification and regression tasks and is especially useful when you want to understand and visualize the decisionmaking process of the model.
- **Gaussian Naive Bayesian**:- Gaussian Naive Bayes is a variant of the Naive Bayes classifier. It assumes that the features follow a Gaussian (normal) distribution. Despite the ”naive” feature independence assumption, it's effective in text classification and continuous data. It calculates the probability that a data point belongs to a particular class based on the likelihood of the features and prior probabilities of the classes, making it a valuable tool for classification tasks.
- **Logistic regression**:- Logistic regression is a classification method that calculates the probability of an event happening based on input variables. It's used for tasks like spam detection or disease prediction. Unlike linear regression, it produces values between 0 and 1, representing probabilities.
- **Random Forest**:- Random Forest is an ensemble learning method that combines multiple decision trees to make more accurate predictions. It works by creating a ”forest” of decision trees and aggregating their outputs. This reduces overfitting and enhances the model's robustness. Random Forest is widely used for classification and regression tasks, offering excellent performance and the ability to handle complex data with high-dimensional feature sets.
- **Extra Trees**:- Extra Trees, short for Extremely Randomized Trees, is an ensemble earning technique similar to Random Forest. It builds multiple decision trees, but with

a key difference: it uses random feature selection and random splitting thresholds at each node. This randomness makes Extra Trees less prone to overfitting and often results in faster training. It's a powerful algorithm for classification and regression tasks, particularly when dealing with noisy or high-dimensional data.

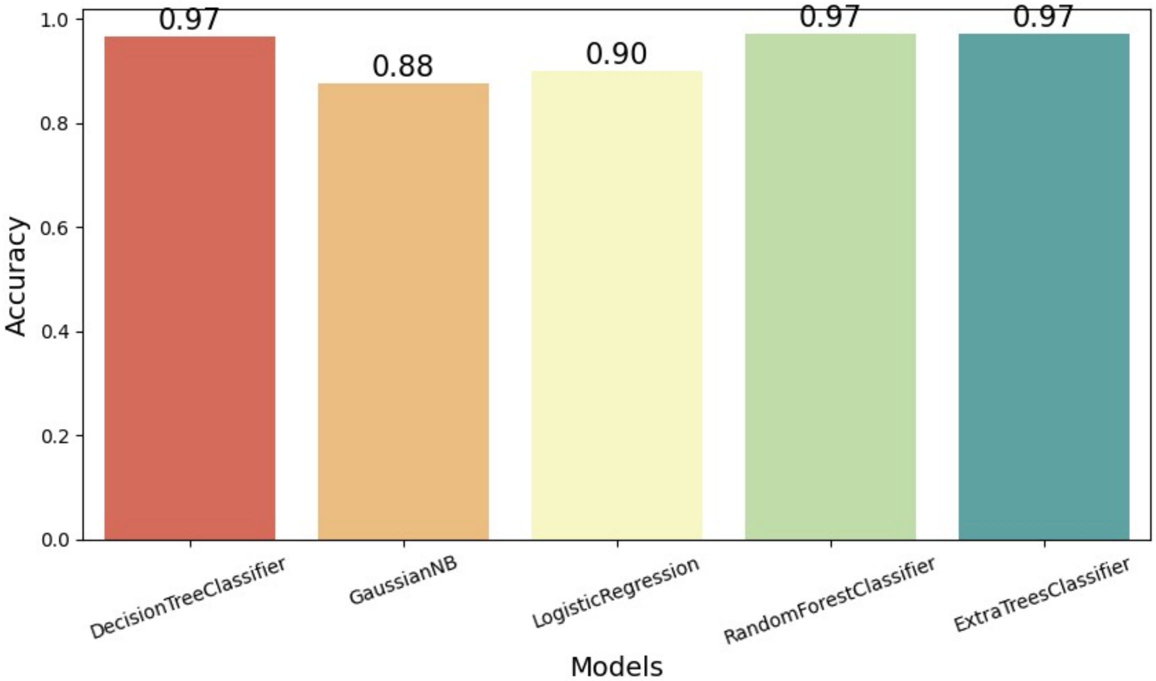


Figure 3.1: Comparative Study of Machine Learning Classifier

To ensure we made an informed choice for our project, we thoroughly examined and compared the accuracy of several machine learning algorithms during our proposed system study. The Decision Tree, Gaussian Naive Bayesian, Logistic Regression, Random Forest, and Extra Tree algorithms were rigorously examined. The Decision Tree achieved an accuracy of 90.53%, Gaussian Naive Bayes achieved an accuracy of 77.86%, Logistic Regression achieved a commendable 91.257% accuracy, Random Forest also delivered 99.70% accuracy, and Extra Tree outperformed the others with an outstanding accuracy of 99.70%.

Several considerations led us to use the Extra Tree algorithm for the project. Initially, its exceptional precision made it the best option for our use case, where dependability and accuracy were crucial. Second, Extra Tree is renowned for its robust prediction powers and capacity to manage intricate, large-scale datasets—a fit that was ideal for the project's requirements along with its high computational speed. To further comprehend the underlying patterns in data, Extra Tree provides exceptional feature selection and model interpretability. Overall, choosing Extra Tree was a wise choice that was motivated by the project's need for dependable outcomes and maximum performance.

3.2 Methodology

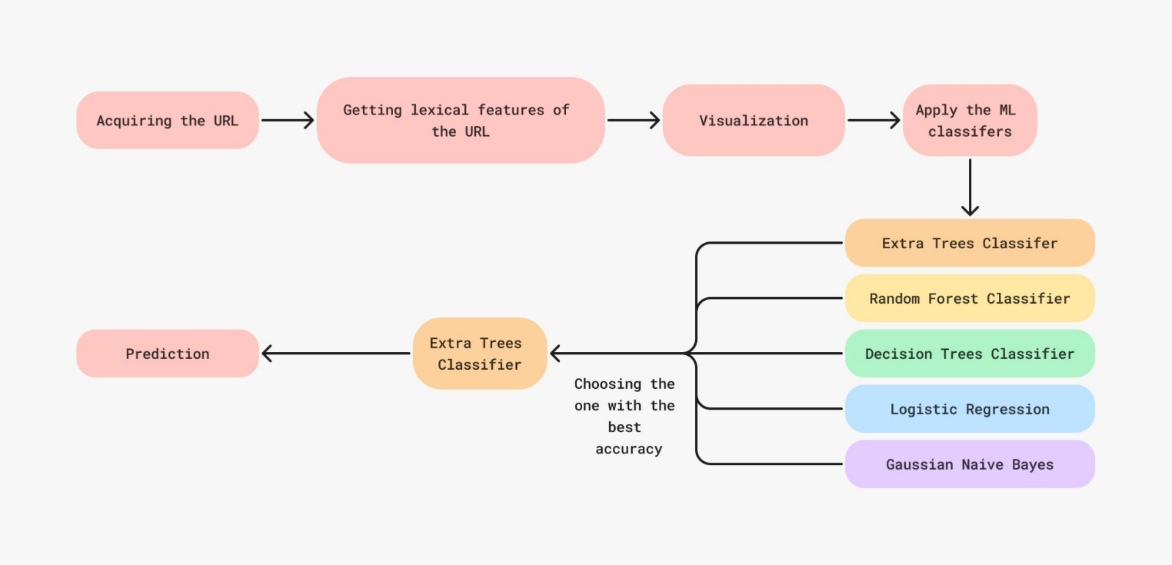


Figure 3.2: Methodology

A QR code, or Quick Response code, is a two-dimensional barcode that stores data. Users can scan QR codes using smartphones or QR code readers to access the information encoded within the code, making it a convenient way to access websites quickly, save contact details, and more. In the application proposed QR codes will be scanned and URLs will be obtained.

A URL, or Uniform Resource Locator, is a web address that specifies the location of a resource on the internet. It consists of different components, including the protocol (e.g., ”http” or ”https”), the domain name (e.g., www.example.com), and the specific path or resource (e.g., /page/index.html) that allows web browsers to retrieve and display web pages and other online content. URLs are essential for navigating the web and accessing websites, files, and online services.

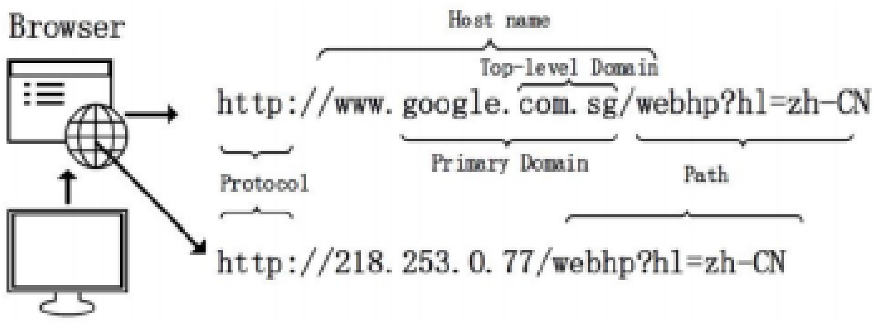

Browser
Host name
Top-level Domain
http://www.google.com.sg/webhp?hl=zh-CN
Protocol
Primary Domain
Path
http://218.253.0.77/webhp?hl=zh-CN

Figure 3.3: URL

After getting the URL all the lexical features are obtained from the URL. These features help in determining whether the URL is malicious or not. The Lexical features obtained are:

1. length of URL
2. length of top-level domain
3. count of ”http”
4. count of percent
5. count of the question ”
6. count of hyphen ”-”
7. length of the hostname
8. count of https
9. count of www
10. count of dot ”.”
11. count of atrate ”@”
12. number of directories
13. number of embedded URL 14. length of First directory
15. count of digits
16. count of letters
17. use of shortening URL services
18. URL containing the IP address

After the lexical features are obtained they are passed to five different ML classifiers which are Decision Tree, Gaussian Naivebayesian, Logistic Regression, Random Forest, and Extra Tree. From the above five classifiers, the classifier which gives us the best accuracy will be chosen.

From Fig3.1 Extra Tree classifier is considered the best classifier as it gives us an accuracy of 0.97 and its computational speed is better as compared to the other classifiers. After the model is trained the user can enter a URL and the model will predict whether the URL is malicious or benign.

Chapter 4

Future Scope

The proposed system has implemented a total of 18 lexical features to enhance the accuracy of our malicious URL detector. Looking ahead, We plan to integrate even more features to ensure the system can provide even more precise results. Right now, the dataset is of approximately 6.51 Lakh URLs. However, there is a need to achieve a balanced ratio between benign and malicious samples.

Regarding the execution environment, a command line interface is used. This choice allows for a more streamlined development process, enabling us to focus on refining the core functionality of the malicious URL Detector.

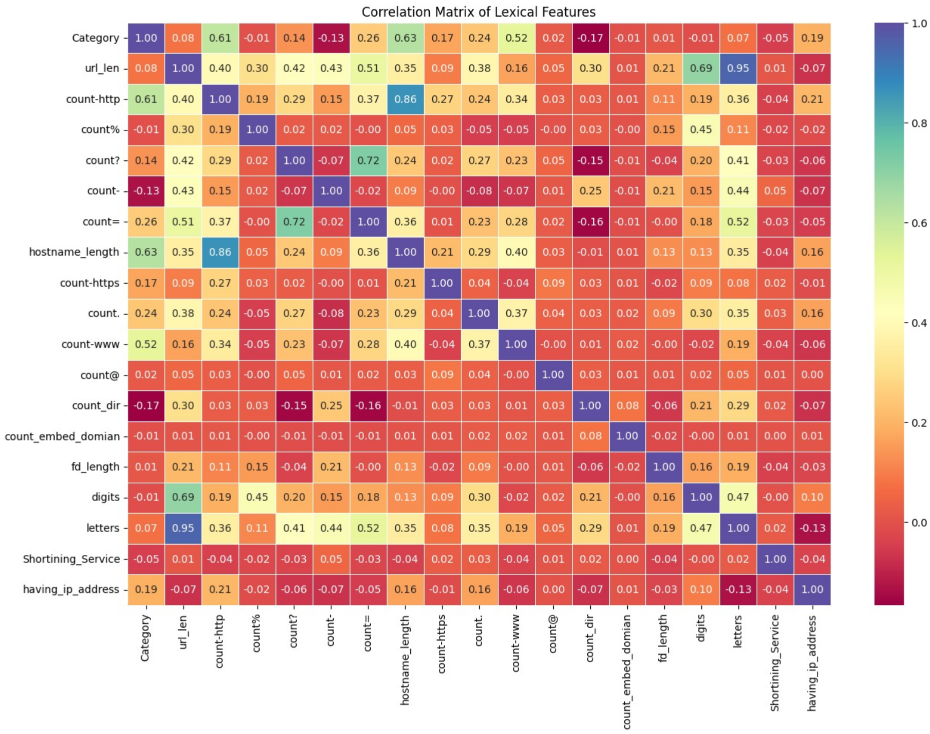
Also this is command line execution. An application will be built. The application will scan the QR code and will identify that the QR code is malicious or benign.

Once the application is up and running, users will be able to effortlessly scan QR codes using their device's camera. Our goal is to make the process as seamless as possible, providing quick results for every scan. If the QR code turns out to be benign, the application will smoothly redirect the user to a specific page, tailored to their needs. On the other hand, if the QR code is flagged as malicious, the application will present the user with options. They can choose to exit the application to stay on the safe side, or they can opt to proceed with caution.

This project is not only ambitious but also crucial in the realm of digital security. The system is continuously dedicated to refining and improving the scanner, ensuring it remains effective in the face of evolving threats. Regular updates and enhancements will be key to maintaining the system's effectiveness over time."

Chapter 5

Results and Analysis



	Category	url_len	count-http	count%	count?	count-	count=	hostname_length	count-https	count.	count-www	count@	count_dir	count_embed_domian	fd_length	digits	letters	Shortining_Service	having_ip_address
Category	1.00	0.08	0.61	-0.01	0.14	-0.13	0.26	0.63	0.17	0.24	0.52	0.02	-0.17	-0.01	0.01	-0.01	0.07	-0.05	0.19
url_len	0.08	1.00	0.40	0.30	0.42	0.43	0.51	0.35	0.09	0.38	0.16	0.05	0.30	0.01	0.21	0.69	0.95	0.01	-0.07
count-http	0.61	0.40	1.00	0.19	0.29	0.15	0.37	0.86	0.27	0.24	0.34	0.03	0.03	0.01	0.11	0.19	0.36	-0.04	0.21
count%	-0.01	0.30	0.19	1.00	0.02	0.02	-0.00	0.05	0.03	-0.05	-0.05	-0.00	0.03	-0.00	0.15	0.45	0.11	-0.02	-0.02
count?	0.14	0.42	0.29	0.02	1.00	-0.07	0.72	0.24	0.02	0.27	0.23	0.05	-0.15	-0.01	-0.04	0.20	0.41	-0.03	-0.06
count-	-0.13	0.43	0.15	0.02	-0.07	1.00	-0.02	0.09	-0.00	-0.08	-0.07	0.01	0.25	-0.01	0.21	0.15	0.44	0.05	-0.07
count=	0.26	0.51	0.37	-0.00	0.72	-0.02	1.00	0.36	0.01	0.23	0.28	0.02	-0.16	-0.01	-0.00	0.18	0.52	-0.03	-0.05
hostname_length	0.63	0.35	0.86	0.05	0.24	0.09	0.36	1.00	0.21	0.29	0.40	0.03	-0.01	0.01	0.13	0.13	0.35	-0.04	0.16
count-https	0.17	0.09	0.27	0.03	0.02	-0.00	0.01	0.21	1.00	0.04	-0.04	0.09	0.03	0.01	-0.02	0.09	0.08	0.02	-0.01
count.	0.24	0.38	0.24	-0.05	0.27	-0.08	0.23	0.29	0.04	1.00	0.37	0.04	0.03	0.02	0.09	0.30	0.35	0.03	0.16
count-www	0.52	0.16	0.34	-0.05	0.23	-0.07	0.28	0.40	-0.04	0.37	1.00	-0.00	0.01	0.02	-0.00	-0.02	0.19	-0.04	-0.06
count@	0.02	0.05	0.03	-0.00	0.05	0.01	0.02	0.03	0.09	0.04	-0.00	1.00	0.03	0.01	0.01	0.02	0.05	0.01	0.00
count_dir	-0.17	0.30	0.03	0.03	-0.15	0.25	-0.16	-0.01	0.03	0.03	0.01	0.03	1.00	0.08	-0.06	0.21	0.29	0.02	-0.07
count_embed_domian	-0.01	0.01	0.01	-0.00	-0.01	-0.01	-0.01	0.01	0.01	0.02	0.02	0.01	0.08	1.00	-0.02	-0.00	0.01	0.00	0.01
fd_length	0.01	0.21	0.11	0.15	-0.04	0.21	-0.00	0.13	-0.02	0.09	-0.00	0.01	-0.06	-0.02	1.00	0.16	0.19	-0.04	-0.03
digits	-0.01	0.69	0.19	0.45	0.20	0.15	0.18	0.13	0.09	0.30	-0.02	0.02	0.21	-0.00	0.16	1.00	0.47	-0.00	0.10
letters	0.07	0.95	0.36	0.11	0.41	0.44	0.52	0.35	0.08	0.35	0.19	0.05	0.29	0.01	0.19	0.47	1.00	0.02	-0.13
Shortining_Service	-0.05	0.01	-0.04	-0.02	-0.03	0.05	-0.03	-0.04	0.02	0.03	-0.04	0.01	0.02	0.00	-0.04	-0.00	0.02	1.00	-0.04
having_ip_address	0.19	-0.07	0.21	-0.02	-0.06	-0.07	-0.05	0.16	-0.01	0.16	-0.06	0.00	-0.07	0.01	-0.03	0.10	-0.13	-0.04	1.00

Figure 5.1: Correlation HeatMap

The heatmap visualizes the correlation matrix of the extracted lexical features used for malicious URL detection. Each cell in the heatmap represents the correlation coefficient between two features, ranging from -1 (strong negative correlation) to +1 (strong positive correlation). Features with high positive or negative correlation are highlighted, helping to identify relationships that may impact the machine learning model's performance.

This visualization aids in understanding feature dependencies, identifying multicollinearity, and selecting the most relevant features for model training. The use of the **Spectral colormap** enhances readability by displaying a gradient of correlation values, while annotations provide precise numeric correlations for detailed analysis.

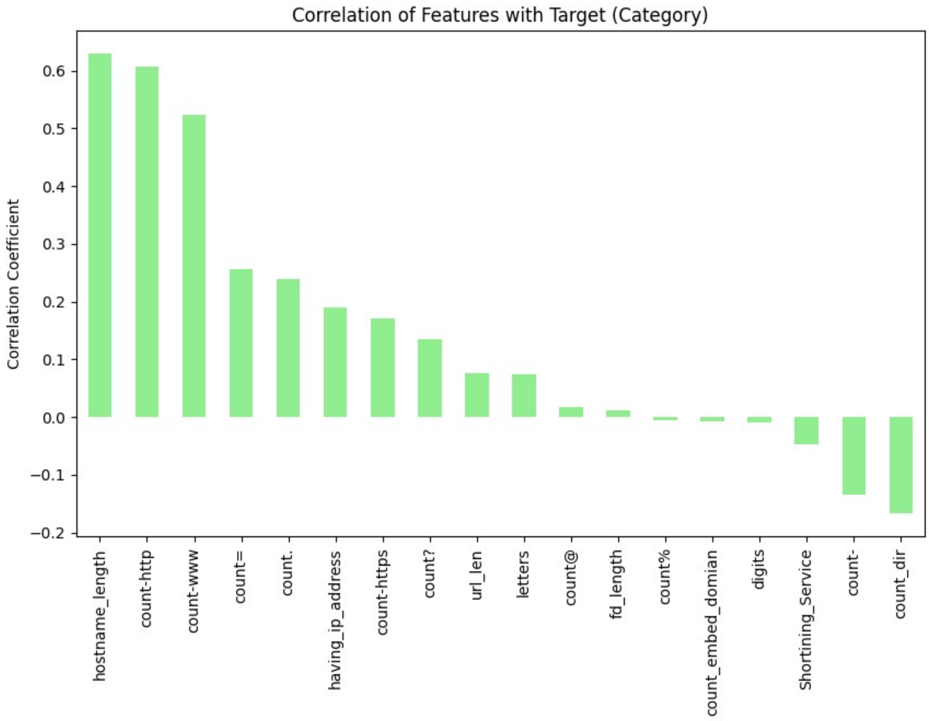
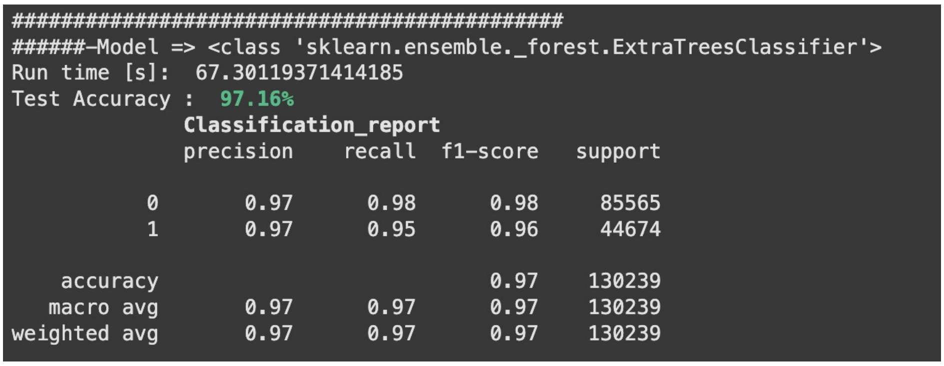


Figure 5.2: Correlation of Features with Target

The bar graph illustrates the **correlation coefficients** between the extracted lexical features and the target variable, **Category** (indicating whether a URL is malicious or benign). Each bar represents the strength and direction of the relationship between a feature and the target, with values ranging from -1 to +1.

Features with higher positive or negative correlations have a stronger predictive relationship with the target, making them potentially more influential in the machine learning model. The bar graph, sorted in descending order, allows for easy identification of the most relevant features. The **light green color** enhances visual clarity, while labeled axes and the title provide context for understanding the importance of each feature in the detection system. This analysis helps prioritize feature selection for improving the model's accuracy and efficiency.



```
###############################################
######-Model => <class 'sklearn.ensemble._forest.ExtraTreesClassifier'>
Run time [s]:  67.30119371414185
Test Accuracy :  97.16%
              Classification_report
              precision    recall  f1-score   support

           0       0.97      0.98      0.98     85565
           1       0.97      0.95      0.96     44674

    accuracy                           0.97    130239
   macro avg       0.97      0.97      0.97    130239
weighted avg       0.97      0.97      0.97    130239
```

Figure 5.3: Extra Trees Performance

The ExtraTreesClassifier achieved excellent performance in the binary classification task, with an accuracy of 97.16% on a dataset of 130,239 samples. The model demonstrated balanced precision (0.97) and recall (0.98 for benign and 0.95 for malicious), resulting in F1-scores of 0.98 for benign and 0.96 for malicious classes. These metrics highlight the model's ability to minimize false positives and effectively identify both classes. The runtime was 67.3 seconds, reflecting the computational demands of processing a large dataset. Overall, the model exhibits high reliability and accuracy, making it well-suited for this classification problem. Optimization for runtime could improve efficiency.

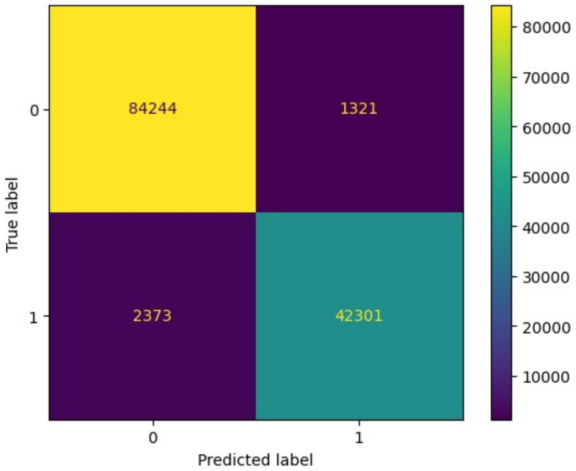
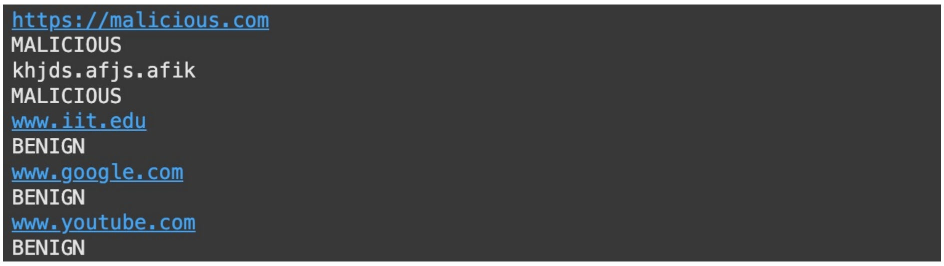


Figure 5.4: Extra Trees Confusion Matrix

The confusion matrix illustrates the performance of the ExtraTreesClassifier on the binary classification task. The model correctly predicted 84,244 benign samples (True Negatives) and 42,301 malicious samples (True Positives). It misclassified 1,321 benign samples as malicious (False Positives) and 2,373 malicious samples as benign (False Negatives). This reflects the model's strong capability to accurately identify both classes, with a slight tendency to misclassify malicious samples. The high numbers of true predictions and relatively low misclassifications align with the overall accuracy of 97.16%, confirming the model’s reliability for the given task. Minor adjustments could further improve its precision.



```
https://malicious.com
MALICIOUS
khjds.afjs.afik
MALICIOUS
www.iit.edu
BENIGN
www.google.com
BENIGN
www.youtube.com
BENIGN
```

Figure 5.5: Prediction

The classification output demonstrates the model’s ability to distinguish between benign and malicious URLs. For example, URLs such as ”https://malicious.com” and ”khjds.afjs.afik” are correctly labeled as malicious, while ”www.iit.edu,” ”www.google.com,” and ”www.youtube.com” are identified as benign. The model consistently evaluates URLs, showcasing its effectiveness in detecting threats.

Chapter 6

Summary

The Malicious URL Detection project aims to develop an effective system for identifying harmful URLs using machine learning techniques. With the rise of cyber threats such as phishing, malware, and data theft, malicious URLs have become a significant concern. This project focuses on creating a machine learning model that can classify URLs as benign or malicious based on features such as domain name, URL length, and query parameters. A diverse dataset of both malicious and benign URLs will be used to train and fine-tune the model, with a target detection accuracy of 95% or higher.

The system will be rigorously evaluated using performance metrics to ensure reliability, and will be regularly updated to address emerging threats. The project also prioritizes **user education**, providing guidelines on how to identify and avoid malicious URLs. By offering an accurate and efficient detection system, the project aims to protect users from online threats, ensuring a safer browsing experience and preventing data theft or financial loss.

Bibliography

[1] Pawar, Atharva, et al. ”Secure QR Code Scanner to Detect Malicious URL using Machine Learning.” 2022 2nd Asian Conference on Innovation in Technology (ASIANCON). IEEE, 2022.

[2] Raja, A. Saleem, R. Vinodini, and A. Kavitha. ”Lexical features based malicious URL detection using machine learning techniques.” Materials Today: Proceedings 47 (2021): 163-166.

[3] AP, Mrs Latha, et al. ”Malicious URL Detection using Logistic Regression.”

[4] Janet, B., and R. Joshua Arul Kumar. ”Malicious URL detection: a comparative study.” 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS). IEEE, 2021.

[5] Do Xuan, Cho, Hoa Dinh Nguyen, and Victor Nikolaevich Tisenko. ”Malicious URL detection based on machine learning.” International Journal of Advanced Computer Science and Applications 11.1 (2020).