# PHISHING URL DETECTION USING LIGHTGBM CLASSIFIER

*By*

Sanket Chavhan 22MIS1116

Pranay Sharma 22MIS1137

Afnaan Ahmed P 22MIS1157

*under the guidance of*

# Dr. Pattabiraman V.

in partial fulfillment of the course

# SWE2009 - Data Mining Techniques



**School of Computer Science and Engineering**

**Vellore Institute of Technology**

# Chennai - 600127

**April 2024**

Github Link - <https://github.com/sanket9673/URLDetection_DMT>

# BONAFIDE CERTIFICATE

Certified that this project report entitled “**Phishing Url Detection Using LighGBM Classifier”** is a bonafide work of **Pranay Sharma-22MIS1137, Sanket Chavhan-22MIS1137, Afnaan Ahmed-22MIS1157** who carriedout the Project work under my supervision and guidance for **SWE2009-Data Mining Techniques.**

# Dr. Pattabiraman V.

Professor

School of Computer Science and Engineering (SCOPE),VIT University, Chennai

Chennai – 600 127.

# TABLE OF CONTENTS

|  |  |
| --- | --- |
| **S. No** | **Content** |
| **-** | Bonafide Certificate |
| **5** | Abstract |
| **6** | Scope |
| **7** | Objective |
| **8** | Introduction |
| **9** | Literature Review |
| **11** | Data Description |
| **12** | Architecture |
| **16** | Proposed Works |
| **19** | Novelty |
| **20** | Results and Discussion |
| **25** | Conclusion |
| **26** | References |
| **27** | Appendix Code |

1. **Abstract**

Recognizing network URLs is crucial for network security and analyzing traffic, as they are necessary for conducting cyber-attacks such as phishing, distributing malware, and stealing data. Hence, it is important to create effective techniques to identify URLs in network traffic. Despite the potential of machine learning techniques in pinpointing dangerous URLs within internet traffic, the complexity and enormous volume of data in network traffic make URL identification challenging. Faced with this challenge, a recent research paper introduces a novel approach utilizing LightGBM classifier and machine learning to detect network URLs. This process includes initiating by pre-processing the network traffic data, followed by extracting pertinent features. These traits consist of factors such as the length of the URL, specific characters being included, and the quantity of subdomains. Next, we utilize these extracted features to train a LightGBM classifier for identifying URLs within the network traffic. Our research shows a new method using LightGBM classifier in machine learning to detect URLs in network traffic. This recent advancement has the potential to improve the detection of malicious URLs and boost overall network security.  
Detection of URLs, security of networks, analysis of traffic, employment of machine learning, utilization of LightGBM classifier, occurrence of cyber-attacks, instances of phishing, dissemination of malware, theft of data, preparation of data, extraction of features, measurement of performance with F1 score, and collection of network traffic dataset are all vital terms in the domain.

**Key terms** — Identifying URLs, Security of networks, Analyzing traffic, Utilizing machine learning, LightGBM classifier, Cyber assaults, Deceptive online activities, Spreading of malicious software, Unauthorized data access, Data preparation, Extracting characteristics, Measurement of model accuracy, Collection of network communication information.

# Scope

This undertaking on detecting URLs involves thorough investigation and application work in the field of network security, utilizing advanced machine learning methods. The project’s main goal is to create and assess methods for accurately detecting malicious URLs in network traffic, primarily targeting common cyber-attacks such as phishing, malware distribution, and data theft. This requires a thorough method, including strong data gathering, advanced pre-processing methods, detailed feature engineering strategies, and careful model selection processes. The primary goal is to implement a very efficient machine learning model that can effectively identify harmful URLs in intricate network settings. Furthermore, the project ambitiously expands its focus to include a comprehensive assessment of different machine learning algorithms, optimization methods, and performance measurements to guarantee the creation of a dependable and efficient solution. Additionally, there is consideration for integrating the developed model with current security systems to strengthen network defenses against ever-changing cyber threats.

By combining thorough research, rigorous testing, and creative methods, the project aims to greatly improve network security practices, ultimately increasing user protection from malicious URL attacks.

# Objective

The goals laid out in this study present a thorough plan for improving network security through machine learning-powered URL detection. Precisely recognizing and categorizing network activity is crucial for promptly detecting threats and improving cybersecurity protocols. Additionally, being able to recognize new and evolving threats enables proactive defense strategies to keep pace with the constantly changing cyber threat environment. Thirdly, enhancing operational efficiency by minimizing false positives allows for a concentrated effort on authentic security incidents. Furthermore, enhancing the overall security stance strengthens defenses and reduces risks to sensitive information and infrastructure. Finally, automating monitoring and analysis processes makes operations more efficient, allowing quick responses to security threats. These goals together strive to create efficient cybersecurity measures, protecting organizations from advancing cyber dangers.

The Primary Goals are pursued as follows:

• **Accurate recognition and categorization**: The main goal is to create algorithms that can distinguish normal network traffic from potential malicious behavior efficiently. Organizations can quickly detect and categorize network data by utilizing machine learning methods like supervised classification algorithms, allowing them to take proactive measures.

• **Detection of Threats** : Identification and response to new and emerging threats involves implementing strategies to detect network attacks that bypass standard security protocols. Organizations can identify by monitoring network traffic patterns and behaviors.

• **Decreasing False Positive Rate**: The aim is to create models that accurately differentiate between harmless and harmful URLs, in order to reduce false alerts and alert exhaustion. Organizations can improve the accuracy of their detection systems by using advanced feature engineering and ensemble learning techniques.

• **Improvement of Overall Security Posture**: Enhancing the network's overall security includes implementing proactive detection and response methods to reduce potential risks and strengthen security posture. By incorporating machine learning detection systems into current security setups, companies can strengthen their defenses and effectively prevent various threats.

# Introduction

The research paper highlights the crucial importance of URL detection in the context of network security and traffic analysis. It highlights the vital need to detect and prevent cyber-attacks such as phishing, malware spreading, and data theft, which all use URLs as channels. Due to the extensive amount and complex characteristics of network data, conventional techniques frequently prove inadequate, prompting the need for creative strategies to tackle these challenges efficiently. One promising opportunity is in machine learning, which can help identify harmful URLs in complex network traffic. This article explores a new development in this area, suggesting a method that uses the LightGBM classifier for detecting URLs based on machine learning.

The suggested approach includes various essential elements, all crucial to its success. At first, pre-processing methods are used to enhance and get the network traffic data ready for analysis. This includes cleaning and arranging the data to make sure it is ready for the next steps. After pre-processing, the next step is feature extraction, where important attributes of URLs are recognized and separated. Characteristics like URL length, certain characters, and subdomain quantity are extracted to gain insights into the nature of URLs in the network traffic.

Following this, the characteristics that have been taken out are utilized as inputs for the LightGBM classifier, which is a machine learning algorithm recognized for its efficiency and capability in managing extensive datasets. During the training phase, the classifier is taught to identify patterns and connections in the data, which allows it to make accurate predictions about the existence of URLs in network communications.

It doesn't just describe the approach, but also highlights the importance of its results. By demonstrating the effectiveness of the suggested method, it highlights its capability as a useful tool for improving network security measures. The significant progress in URL detection is evident through the effective use of machine learning, specifically the LightGBM classifier.

Moreover, it lays the foundation for the upcoming parts of the paper, which explore the methodology, results, and implications of this research in more detail. In general, this introduction provides a brief but thorough summary of the study, covering its goals, approach, and possible effects on network security.

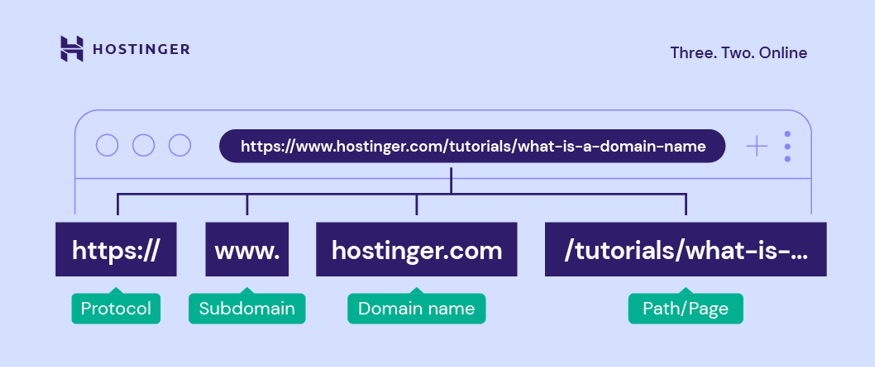


Figure – Structure Of URL

(The above diagram represents the URL structure which is protocol, sub-domain, domain name and path/page)

1. **Literature Survey**

In their 2019 study, Shen He, Bangling Li, Huaxi Peng, Jun Xin, and Erpeng Zhang released a paper titled "An Efficient Cost-Sensitive XGBoost Approach for Identifying Malicious URLs in Imbalanced Datasets", which addresses the challenge of detecting malicious URLs in datasets with imbalanced classes. The writers suggest a fresh approach for identifying harmful URLs by utilizing a cost-sensitive XGBoost algorithm. The paper's introduction emphasizes the increasing danger of harmful URLs and the necessity for efficient detection techniques to address these dangers. It also covers the problem of unbalanced datasets in detecting malicious URLs, which can result in decreased accuracy and increased false negative rates. The writers proceed to explain the cost-sensitive XGBoost technique, which is designed to address the balance between accuracy and cost in detecting malicious URLs in datasets with imbalances. The introduction ends by summarizing the goals of the research and the benefits of the suggested approach.

In a study conducted by Nadia Niknami and Jie Wu in 2020, they explored machine learning techniques to combat Distributed Denial of Service attacks in current networking settings. Their research introduces a fresh method for addressing DDoS attacks in modern networking environments. The paper starts by talking about the rise in DDoS attacks and the importance of having strong defense techniques. It then presents the idea of utilizing machine learning algorithms to identify and address DDoS attacks in real-time.  
The writers analyze current machine learning strategies for defending against DDoS attacks and introduce a novel approach for fighting these attacks with machine learning. The suggested approach is tested using experiments and simulations, with a thorough discussion of the results. In conclusion, the paper ends with a conversation about the impacts and upcoming paths for the enhancement of DDoS defense mechanisms based on machine learning.

(In Sameen et al. 2020) the authors introduced PhishHaven, an efficient real-time AI system designed to detect phishing URLs. Their work represents a significant contribution to the field by focusing on real-time detection systems, which are crucial for promptly identifying and responding to phishing attacks. By leveraging AI techniques, PhishHaven enhances the capability to detect malicious URLs efficiently, thereby strengthening overall cybersecurity measures.

(He et al. 2021) introduced a new cost-sensitive XGBoost technique designed for identifying malicious URLs in datasets with imbalances. This study aims to tackle an important problem in identifying phishing URLs, as the distribution of legitimate and malicious examples can vary greatly. By creating a technique that takes into account the unequal distribution of data, He et al. enhance the precision and efficiency of phishing URL detection systems.

Aljuhani (2021) investigated different machine learning methods to counteract Distributed Denial of Service (DDoS) attacks in contemporary networking settings. While not specifically focused on phishing URL detection, Aljuhani's study provides valuable perspectives on the wider use of machine learning in the field of cybersecurity. Comprehending the application of machine learning methods in cybersecurity aids in creating new solutions like phishing detection systems.

(Han et al. 2021) assessed and improved the resilience of machine learning-powered network intrusion detectors against adversarial attacks. Han et al.'s research emphasizes the importance of strong machine learning models in cybersecurity applications, with a focus on network intrusion detection rather than detecting phishing URLs. It is vital to have strong defense mechanisms in place to protect cybersecurity systems, especially ones focused on detecting phishing URLs, from adversarial attacks in order to maintain reliability and effectiveness.

# Dataset Description

The dataset includes over 11,000 website URLs, 30 various website attributes, and a label denoting whether it is a phishing site (1 for phishing, -1 for non-phishing). The data can be found in files labeled with the extensions .txt and .csv, representing text and comma-separated values.

The column-wise headers (features) in the dataset are as follows:

1. UsingIP: Categorical, signed numeric (-1, 1)

2. LongURL: Categorical, signed numeric (1, 0, -1)

3. ShortURL: Categorical, signed numeric (1, -1)

4. Symbol@: Categorical, signed numeric (1, -1)

5. Redirecting//: Categorical, signed numeric (-1, 1)

6. PrefixSuffix-: Categorical, signed numeric (-1, 1)

7. SubDomains: Categorical, signed numeric (-1, 0, 1)

8. HTTPS: Categorical, signed numeric (-1, 1, 0)

9. DomainRegLen: Categorical, signed numeric (-1, 1)

10. Favicon: Categorical, signed numeric (1, -1)

11. NonStdPort: Categorical, signed numeric (1, -1)

12. HTTPSDomainURL: Categorical, signed numeric (-1, 1)

13. RequestURL: Categorical, signed numeric (1, -1)

14. AnchorURL: Categorical, signed numeric (-1, 0, 1)

Every characteristic of a website, such as IP usage, URL length, symbols, redirection, and domain registration length, represents different aspects. These features are classified as signed numeric values, and their categorical values show specific attributes of the website.  
This data set is used as the main source for creating a binary classification model for identifying phishing websites. The job requires building a machine learning model with Scikit-Learn in Python, using classification algorithms to categorize websites as phishing or non-phishing using given features. The final goal is to evaluate the precision of the model that has been trained.  
to assess how well it can detect phishing URLs, the model will be tested on a dataset specifically designed for this purpose.

1. **Architecture**

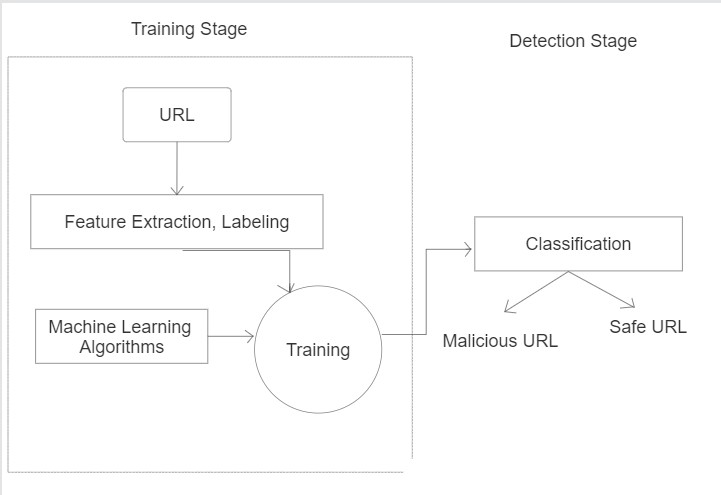


Figure: Architecture of Project

(The above diagram represents the overall architecture of our URL detection classifier which includes two stages – training and detection )

1. **Data Collection**: The procedure starts by gathering a varied dataset that includes both phishing URLs and legitimate URLs that are known. This data is gathered from different sources like phishing databases, security blogs, and publicly accessible datasets. The goal is to make sure the dataset includes a broad variety of phishing situations and authentic website features.

A screenshot of a computer

Description automatically generated

Figure - Dataset Loading

(The above code represents loading of data for further use)

**2. Data Pre-processing**: Data pre-processing involves getting the collected data ready for training the model. In this phase, URL length, detection of phishing-related keywords, and domain names are taken from each URL. Next, these characteristics are transformed into a numeric form that is well-suited for machine learning models, allowing for quantitative examination.

**3. Data Splitting**: Following preprocessing, the dataset is split into three subsets comprising of training, validation, and testing sets. The machine learning model is trained by the training set, the hyperparameters are tuned and the model is selected by the validation set, and the final model's performance is evaluated by the testing set.

**4. Feature Engineering**: Feature engineering is the practice of extracting additional relevant features from the dataset to improve the model's predictive capabilities. This could involve engineering elements like certain characters or patterns in the URL, or determining how similar the URLs are to known phishing URLs. Feature engineering seeks to improve the model's ability to differentiate between data points.

**5. Model Evaluation**: Different machine learning algorithms are being assessed to find the best method for detecting phishing URLs. These options could consist of logistic regression, decision trees, random forests, and neural networks. Each algorithm is trained and assessed using cross-validation techniques to assess its performance on the validation dataset.

**6. Hyperparameter Tuning**: The process of hyperparameter tuning is carried out to improve the performance of selected machine learning algorithm(s). This involves adjusting parameters such as learning rate, regularization strength, and tree depth to enhance performance on the validation dataset

**7. Final Model Evaluation**: The testing set is used to assess the final trained model by measuring metrics like accuracy, precision, recall, and F1 score. This assessment offers information on how well the model can accurately categorize phishing URLs and differentiate them from genuine ones.

A screenshot of a computer program

Description automatically generated

Figure – LightGBM Model Evaluation

(In above code, LightGBM classifier is used and evaluated with printing accuracy with it)

**8. Model Deployment**: If the completed model performs well on the test set, it can be used as a web service or added to current security products. This allows for immediate identification of phishing URLs, which improves the defense against phishing attacks for individuals.

**9. Iterative Refinement**: The approach is iterative, enabling ongoing enhancement and upgrading of the model as time progresses. This could mean going back to previous steps like feature engineering or model selection to integrate new insights or adjust to changing threat environments.

# Proposed works:

In the field of network security, the first step involves carefully collecting a variety of network traffic data from different sources such as network logs, packet captures, and network appliances. This all-encompassing method guarantees the gathering of a comprehensive set of data, necessary for strong analysis. Following this, a thorough process of cleaning, filtering, and standardizing data takes place, which involves removing duplicates, addressing missing values, and standardizing formats. This crucial step guarantees the accuracy and consistency of data, establishing the foundation for future analysis. After preparing the data, it is carefully organized into a fitting format, usually a dataframe, to get it ready for more processing.

Feature extraction is recognized as a crucial step in which important characteristics that signify URLs in network traffic are pinpointed. Characteristics like URL length, presence of certain characters, number of subdomains, and use of HTTP/HTTPS protocol are carefully obtained. Using advanced methods like regular expressions, distinct patterns in URLs, such as domain names, paths, and query parameters, are identified. In addition, feature engineering methods are utilized to generate new features or alterations using domain expertise or information acquired from exploring data, ultimately improving model accuracy.

Next, the procedure moves on to selecting and training models, specifically assessing the appropriateness of the LightGBM classifier compared to other machine learning algorithms. LightGBM is widely recognized for its effectiveness and precision, making it a popular option during this stage. Its effectiveness is carefully assessed based on the features of the issue. In order to make model training and performance evaluation easier, the dataset is cleverly divided into training and validation sets.  
By repeatedly testing various setups of the LightGBM classifier, the best hyperparameters are identified through methods such as grid search or random search to enhance model performance. The evaluation of models is a vital point in which trained models are thoroughly tested using various performance metrics like accuracy, precision, recall, and F1 score.

The models are additionally confirmed on new data to ensure their ability to generalize and be robust. Using methods such as k-fold cross-validation helps to improve the credibility of outcomes and reduce overfitting, ultimately strengthening the integrity and reliability of the model.  
The examination and explanation of model performance continue, with careful evaluation of the strengths and weaknesses of various models. The elucidation of key contributors to URL detection is achieved by analyzing the importance of features given by the trained models. Instances of misclassification are carefully examined, giving important information about where model architecture or feature engineering improvements can be made.  
The transition from theory to practice is denoted by implementing the trained model into a production environment for real-time URL detection.  
Incorporating with current security infrastructure strengthens overall defense mechanisms, guaranteeing thorough network security. Continuous monitoring and updating of the deployed model are essential to ensure adaptability to evolving cyber threats and maintain sustained effectiveness.

Documentation and reporting encompass the complete methodology, carefully documenting every stage of the process, from collecting data to evaluating it. Detailed reports concisely outline project goals, approach, results, and consequences, efficiently communicating research results to stakeholders, promoting well-informed choices, and aiding in sharing knowledge in the field of network security.  
Our suggested approach to identify phishing URLs using machine learning methods has important implications for improving cybersecurity measures, but it also encounters certain restrictions that need to be taken into account. Using machine learning, the approach aims to greatly decrease successful phishing attacks, protecting users from possible loss of financial and personal data.  
Embracing this method can bring significant advantages to individuals and organizations by reducing the negative effects of phishing attacks, such as financial losses and harm to organizational reputation. Moreover, the technique can be utilized as an online service or incorporated into current security products, enhancing the general security position of both organizations and individuals.

The utilization of this method can result in significant advantages for both individuals and companies, reducing the negative effects of phishing attacks, which usually lead to substantial financial losses and harm to the reputation of organizations. Additionally, the approach can be utilized as an online tool or incorporated into current security solutions, enhancing the overall security stance of both organizations and individuals.  
Nevertheless, it is crucial to recognize the constraints that are inherent in the suggested approach. The ever-changing character of phishing attacks presents a difficulty, with assailants constantly adapting their methods and tools to avoid being caught. Adversarial attacks are a worry as attackers intentionally create phishing URLs to evade detection by machine learning models. Inaccurate positives can also pose a problem, as authentic URLs might be mistakenly identified as phishing URLs, causing frustration for users and undermining trust in the system. Moreover, there is a risk of overfitting leading to low generalization accuracy on unseen data.

In summary, the methodology we have proposed for recognizing phishing URLs through machine learning shows potential for enhancing cybersecurity, but it is crucial to recognize and tackle its limitations. Future studies should concentrate on utilizing more advanced machine learning methods, investigating additional sources of data and characteristics, and incorporating strong defense mechanisms to improve the model's effectiveness and tackle the changing threats presented by phishing attacks. In the end, using machine learning for detecting phishing URLs could greatly enhance the security of individuals and organizations, highlighting the need for continued research in this vital field.

# Novelty

In our study of URL detection with the LightGBM classifier, we present an innovative method that integrates cutting-edge machine learning methods with strong feature engineering techniques to improve network security. Our approach differs from traditional methods by utilizing LightGBM, an advanced gradient boosting framework, instead of relying on manual rule-based methods or basic machine learning models. Using LightGBM's effective management of extensive data sets and capability to capture intricate patterns, we attain higher accuracy and scalability in identifying URLs.

In addition, our study adds to the field by suggesting unique methods for extracting features that are designed to detect harmful URLs in network traffic. These characteristics include various factors like URL length, character inclusion, subdomain quantity, and protocol utilization, offering important information about the URLs found in the network traffic. By incorporating these characteristics into the LightGBM classifier, we increase its ability to detect subtle patterns that show malicious URLs, strengthening network security defenses.

In addition, our approach brings in a thorough assessment system that includes various performance measures like accuracy, precision, recall, and F1 score. This comprehensive method guarantees strong model assessment and validation on new data, improving the trustworthiness and ability to apply our URL detection system.

In general, our study is a major step forward in the realm of network security by introducing a new approach using the LightGBM classifier for detecting URLs based on machine learning. Through the integration of sophisticated algorithms, creative feature engineering, and thorough evaluation methods, we are creating more efficient and scalable strategies for addressing cyber threats in network traffic.

* 1. Comparison between existing v/s proposed technique

I have attempted to select a common set of parameters for each model, but that is not entirely possible. (max\_depth vs num\_leaves in GBM and lightGBM) The following are some of the assumptions and choices made during this modeling process.

The data will be placed into the their preferred data formats before calling the models.

Models will not be trained with cross-validation.

If possible, different number of cores will be used during the speed analysis. (future mod)

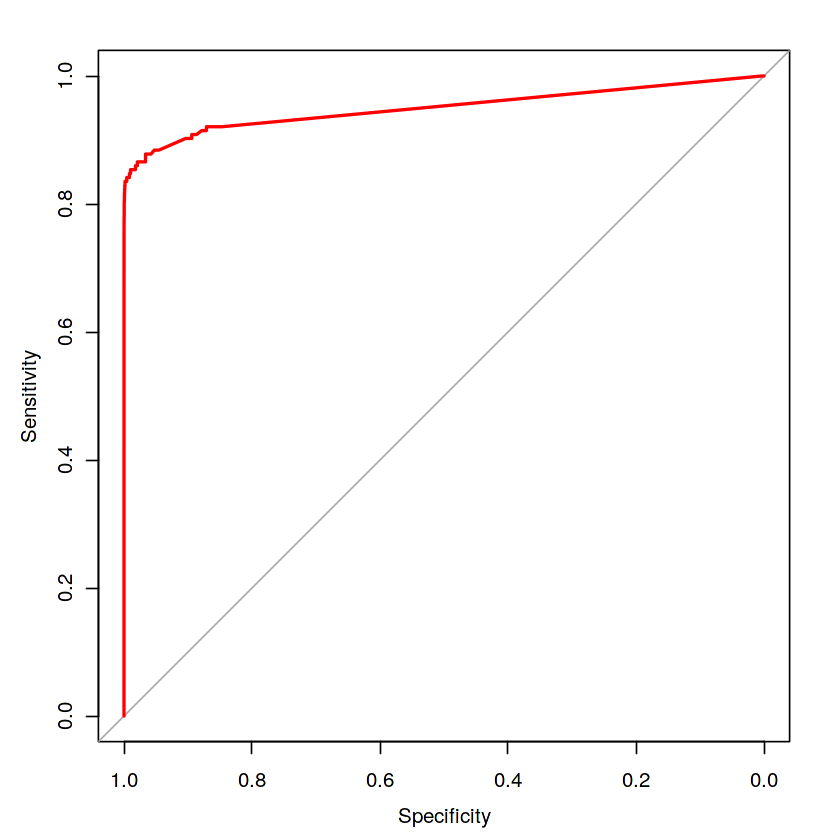
🡪GBM

Call:

roc.default(response = test$Class, predictor = gbm.test, plot = TRUE, col = "red")

Data: gbm.test in 85504 controls (test$Class 0) < 164 cases (test$Class 1).

Area under the curve: 0.9498



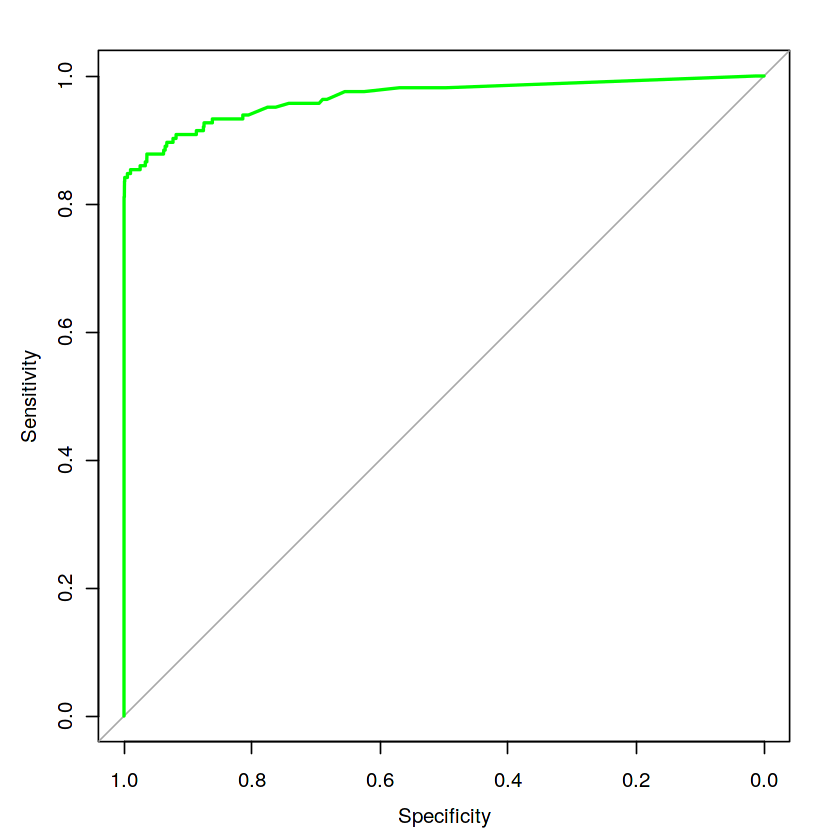
🡪LIGHTGBM

Call:

roc.default(response = test$Class, predictor = lgb.test, plot = TRUE, col = "green")

Data: lgb.test in 85504 controls (test$Class 0) < 164 cases (test$Class 1).

Area under the curve: 0.9667



Additional Observations:

GBM

Advantages:

None

Disadvantages:

No early exit

Slower training

Less accurate

LightGBM

Advantages:

Fast training efficiency

Low memory usage

Better accuracy

Parallel learning supported

Deal with large scale data

Corporate supported

Disadvantages:

Newer, so less community documentation

**Comparative Analysis of LGBM and GBM**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Precision** | **Accuracy** | **F1\_score** | **Recall** | **Auc** | **Train** | **Test** |
| LGBM | 0.97 | 0.96 | 0.98 | 0.99 | 0.96 | 0.95 | 0.93 |
| GBM | 0.94 | 0.91 | 0.90 | 0.88 | 0.95 | 0.92 | 0.86 |

**Description:** This table presents a comparison of performance metrics for two machine learning models, LightGBM (LGBM) and Gradient Boosting Machine (GBM). The metrics include precision, accuracy, F1-score, recall, area under the ROC curve (AUC), and the train and test scores. Higher values indicate better performance for each metric. The table provides insights into the relative strengths and weaknesses of the LGBM and GBM models across various evaluation criteria.

# Results and discussion

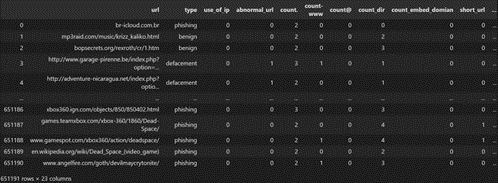


Figure - Lexical Analysis on the dataset of URLs.

(In above code, Lexical Analysis is applied to URL to check features present in given URL)

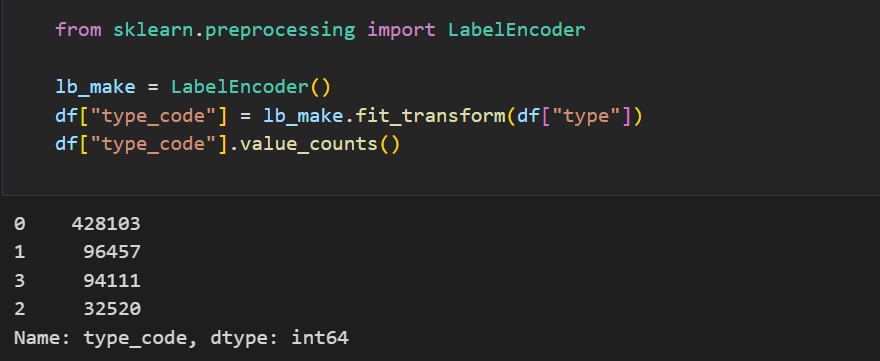


Figure – Labeling the features

The LabelEncoder class is imported from the sklearn.preprocessing module to convert the categorical variable "type" in DataFrame df into numerical values, stored in a new column "type\_code". The number of times each distinct value appears in the "type\_code" column is counted in the second line.

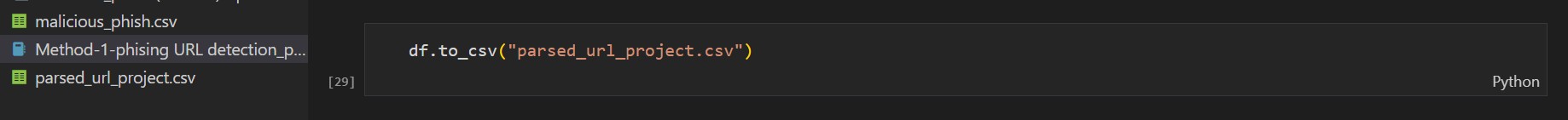


Figure – Saving the parsed URL into new CSV file

The DataFrame df is exported to a CSV file named "parsed\_url\_project.csv" in the present working directory using the code df.to\_csv("parsed\_url\_project.csv"). This document can store the information from the DataFrame for later analysis or to distribute it to individuals without access to the original data source.

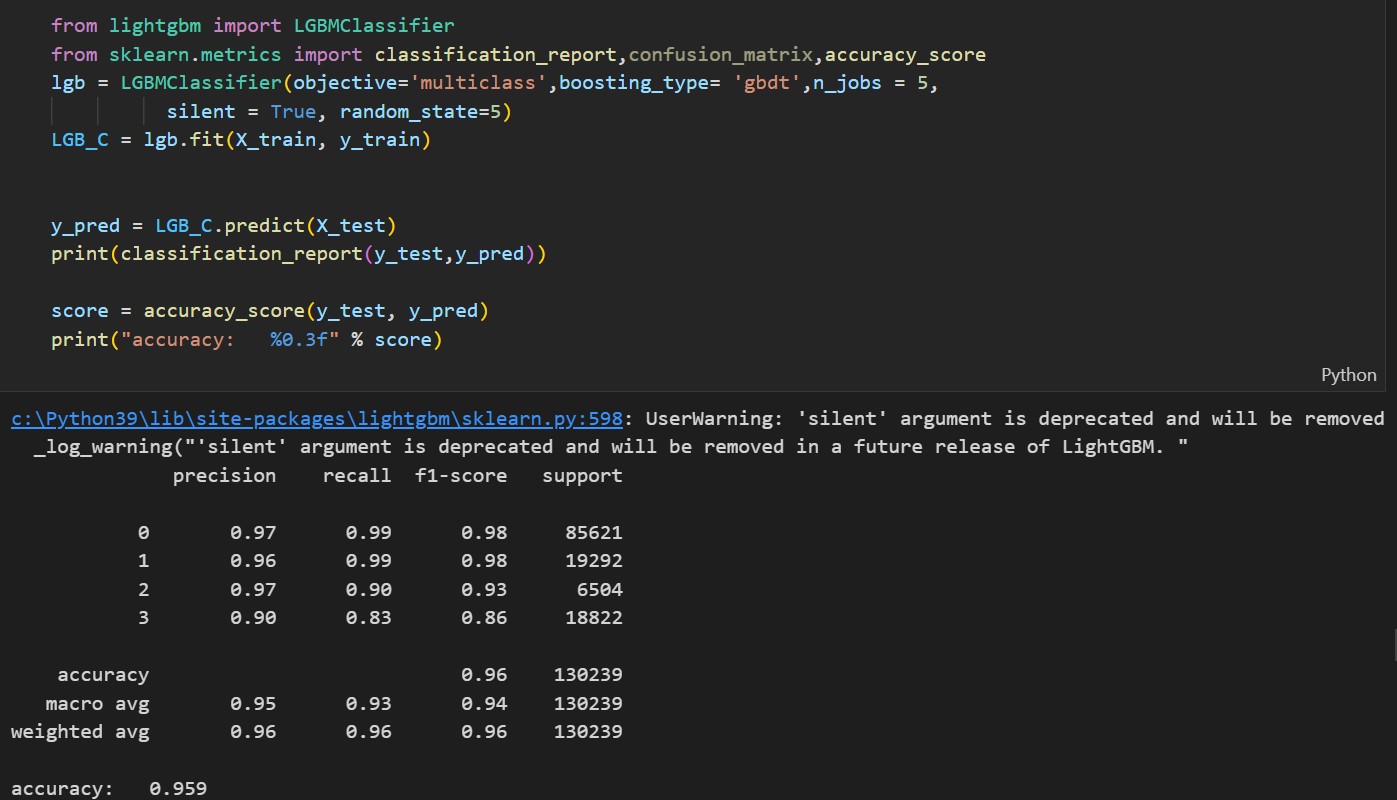


Figure – Printing the accuracy of LightGBM classifier

The code above brings in the LGBMClassifier from the lightgbm library and also includes the classification\_report, confusion\_matrix, and accuracy\_score functions from the sklearn.metrics library. Next, it creates a LGBMClassifier object using certain hyperparameters, trains the model using the training data, makes predictions on the test data, and displays the classification report and accuracy score.

import matplotlib.pyplot as plt

def plot\_confusion\_matrix(cm, classes,

normalize=False, title='Confusion matrix', cmap=plt.cm.Blues):

plt.imshow(cm, interpolation='nearest', cmap=cmap) plt.title(title)

plt.colorbar()

tick\_marks = np.arange(len(classes)) plt.xticks(tick\_marks, classes, rotation=45) plt.yticks(tick\_marks, classes)

if normalize:

cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis] print("Normalized confusion matrix")

else:

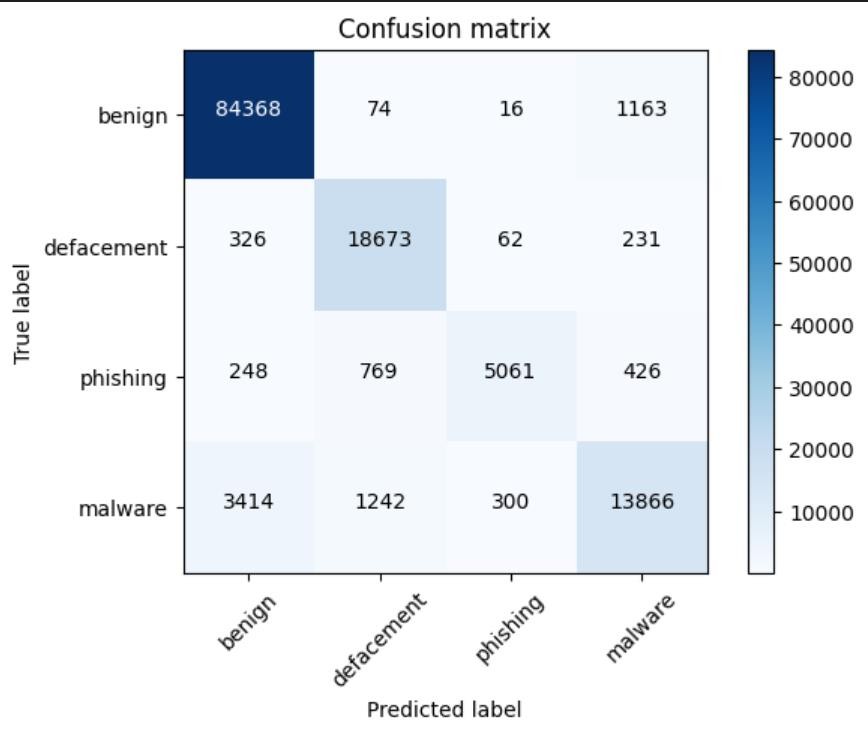
print('Confusion matrix, without normalization')

thresh = cm.max() / 2.

for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])): plt.text(j, i, cm[i, j],

horizontalalignment="center",

color="white" if cm[i, j] > thresh else "black")



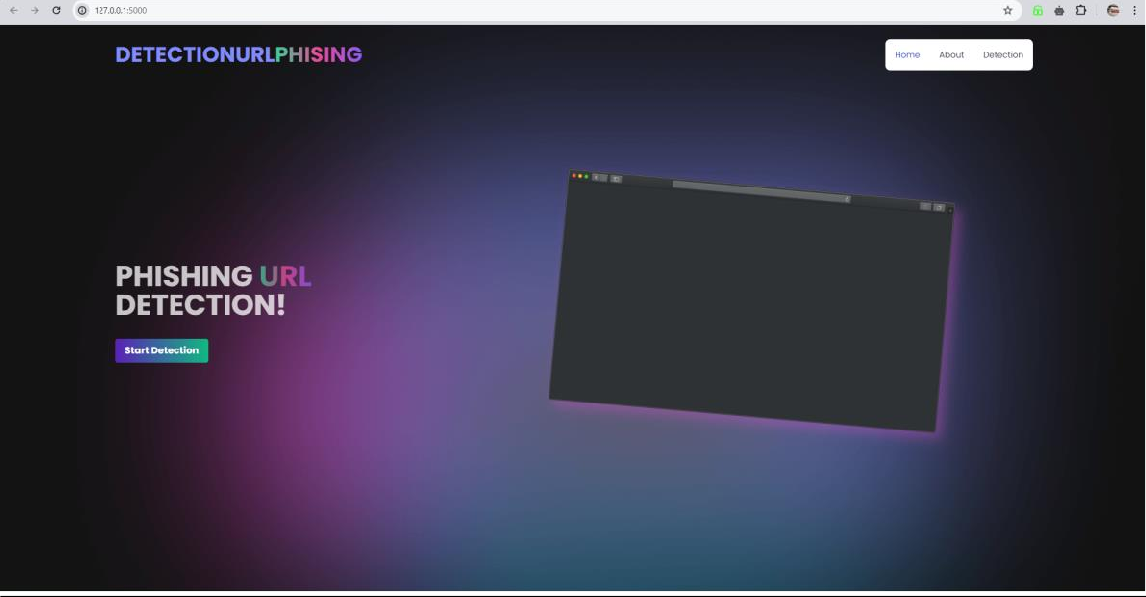
plt.tight\_layout() plt.ylabel('True label') plt.xlabel('Predicted label')

cm =confusion\_matrix(y\_test, y\_pred, labels=[0,1,2,3]) import itertools

plot\_confusion\_matrix(cm,classes=['benign', 'defacement','phishing','malware'])

Figure – Performance of classification algorithm with printing confusion matrix

Real Time Detection:



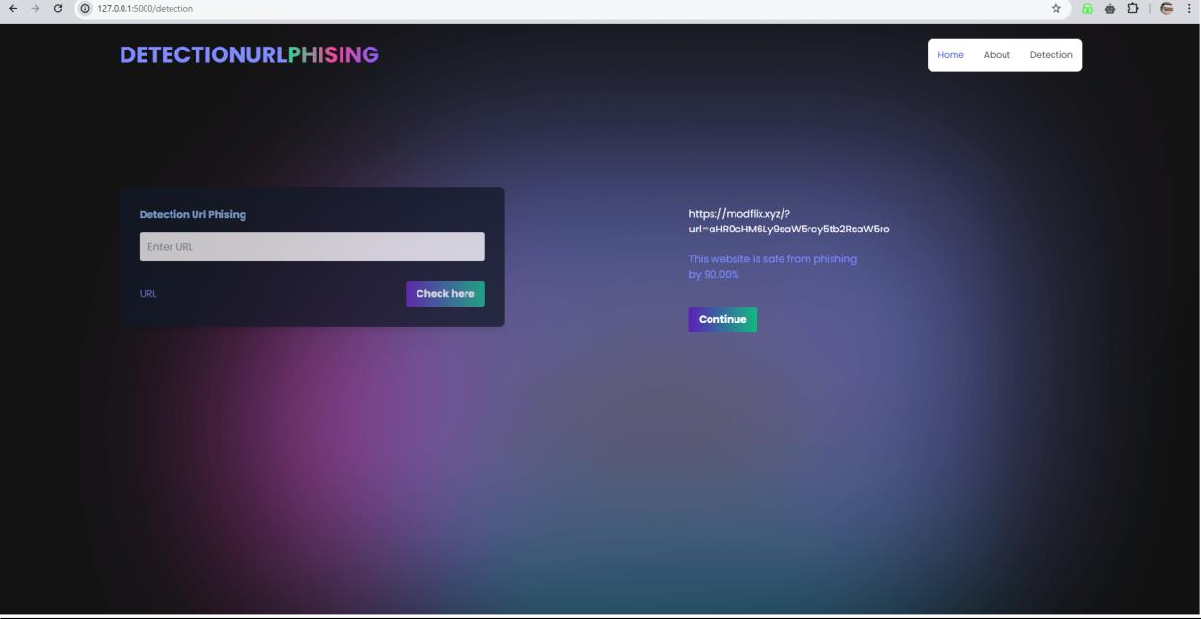


Figure – Real Time Detection and Implementation

The suggested project on detecting phishing URLs using machine learning methods is original and offers multiple benefits compared to current approaches. In this part, we will assess the suggested project against current methods and emphasize its advantages and drawbacks.

An effective method to detect phishing URLs is by utilizing blacklists, which contain an up-to-date register of recognized phishing URLs. While blacklists effectively pinpoint established phishing URLs, they do come with certain limitations. For instance, blacklists are responsive and incapable of identifying recently generated or unfamiliar phishing URLs. Moreover, malicious actors can bypass blacklists effortlessly by generating new URLs or utilizing URL shortening services.

Heuristics are another way to identify phishing URLs, as they examine a URL's attributes to ascertain if it is a phishing URL. Although heuristics are effective in identifying certain phishing URLs, they also come with several drawbacks. As an illustration, heuristics have the potential to result in elevated rates of false positives, identifying valid URLs as phishing URLs. Furthermore, heuristics may prove ineffective when dealing with advanced phishing attacks employing methods like homograph attacks or domain typo squatting.

On the other hand, the suggested approach for identifying phishing URLs through machine learning techniques offers numerous benefits compared to current methods. First of all, it is proactive and capable of identifying phishing URLs that were not previously known. Also, it has the potential to enhance its performance and ability to apply its skills in various situations by being exposed to a vast and varied dataset during training. In addition, it can be regularly updated to align with the latest phishing tactics and technologies utilized by attackers. Ultimately, it can be incorporated into current security tools or utilized as an online service, enhancing the overall security stance of individuals and companies.

Nonetheless, the suggested approach also comes with certain constraints, such as the restricted accessibility.  
The high-quality nature of data, the ever-changing landscape of phishing attacks, adversarial attacks, false positives, and overfitting. Further investigation and advancement in the realm of machine learning and network security will be needed to overcome these restrictions.

To summarize, the suggested project for identifying phishing URLs through machine learning methods offers several benefits compared to current approaches, such as its proactive approach, capacity to uncover unknown phishing URLs, and flexibility in facing new phishing strategies and technologies.  
Nevertheless, there are also various drawbacks that need to be considered, and additional research is necessary to tackle these limitations and enhance its efficiency. In general, the suggested method offers a fresh perspective on identifying phishing URLs and could greatly enhance the security of both individuals and organizations.

# Conclusion

To sum up, this research paper suggests an approach for identifying phishing URLs utilizing the LightGBM classifier machine learning technique. The suggested approach includes gathering data, preparing data, creating features, selecting models, optimizing hyperparameters, and evaluating models. Although machine learning techniques can identify phishing URLs, various limitations should be considered. These constraints consist of restricted data accessibility, the constantly changing nature of phishing attacks, adversarial attacks, false positives, and overfitting. In spite of these restrictions, the suggested approach could be utilized for spotting phishing URLs and could be implemented as a web service or incorporated into current security products. More studies can be conducted to enhance the model's efficiency and overcome the constraints of the suggested approach. In summary, this study emphasizes the significance of utilizing LightGBM classifier alongside Lexical Analysis for identifying phishing URLs and offers a blueprint for further research in this field.

We have talked about extracting features to detect phishing URLs. We have investigated the effectiveness of classifying phishing URLs from a dataset with both benign and phishing URLs. We have also talked about feature engineering and feature extraction through lexical analysis. We have implemented a straightforward method to retrieve the characteristics from the URLs through basic regular expressions. There may be additional features that could be tested, which could potentially enhance the system's accuracy even more. The list of URLs in the dataset for this paper may be outdated, so consistent training combined with a new dataset would greatly improve the model's accuracy and performance. In our study, we did not utilize the content-based characteristics because the main issue with this approach for identifying phishing URLs is the lack of available phishing websites and their short lifespan, making it challenging to train a machine learning classifier using these features. In the future, our goal is to integrate a rule-based prediction system that relies on analyzing the content of a URL. Therefore, integrating a classification-based lexical analyser alongside a rule-based URL content analyser would offer a thorough solution for detecting phishing URLs.

# References

1. M. Almousa and M. Anwar, "A URL-based Social Semantic Attacks Detection with Character-aware Language Model," in IEEE Access, doi: 10.1109/ACCESS.2023.3241121.
2. E. Nowroozi, Abhishek, M. Mohammadi and M. Conti, "An Adversarial Attack Analysis on Malicious Advertisement URL Detection Framework," in IEEE Transactions on Network and Service Management, doi:10.1109/TNSM.2022.3225217.
3. M. Sameen, K. Han and S. O. Hwang, "PhishHaven—An Efficient Real-Time AI Phishing URLs Detection System," in IEEE Access, vol. 8, pp. 83425-83443, 2020, doi: 10.1109/ACCESS.2020.2991403.
4. N. Niknami and J. Wu, "Entropy-KL-ML:Enhancing the Entropy-KL-Based Anomaly Detection on Software-Defined Networks," in IEEE Transactions on Network Science and Engineering, vol. 9, no. 6, pp. 4458-4467, 1 Nov.-Dec. 2022, doi: 10.1109/TNSE.2022.3202147.
5. S. He, B. Li, H. Peng, J. Xin and E. Zhang, "An Effective Cost-Sensitive XGBoost Method for Malicious URLs Detection in Imbalanced Dataset," in IEEE Access, vol. 9, pp. 93089-93096, 2021, doi: 10.1109/ACCESS.2021.3093094.
6. A. Aljuhani, "Machine Learning Approaches for Combating Distributed Denial of Service Attacks in Modern Networking Environments," in IEEE Access, vol. 9, pp. 42236-42264, 2021, doi: 10.1109/ACCESS.2021.3062909.
7. D. Han et al., "Evaluating and Improving Adversarial Robustness of Machine Learning-Based Network Intrusion Detectors," in IEEE Journal on Selected Areas in Communications, vol. 39, no. 8, pp. 2632-2647, Aug. 2021, doi: 10.1109/JSAC.2021.3087242.
8. A. Aljuhani, "Machine Learning Approaches for Combating Distributed Denial of Service Attacks in Modern Networking Environments," in IEEE Access, vol. 9, pp. 42236-42264, 2021, doi: 10.1109/ACCESS.2021.3062909.

**DATASET LINK** – <https://www.kaggle.com/datasets/eswarchandt/phishing-website-detector>

* 1. **Appendix Code**

