Project report on

**Security in social networks( phishing URL predictor)\***

**By**

Prem Ramesh Satpute

**UNDER THE SUPERVISON OF**

Qriocity



**Aim :**

The aim of this project is to develop a machine learning model capable of accurately predicting phishing URLs to enhance security measures in social networks.

**Objective :**

A phishing website is a common social engineering method that mimics trustful uniform resource locators (URLs) and webpages. The objective of this notebook is to collect data & extract the selective features form the URLs.

1. To collect and preprocess a comprehensive dataset of URLs, including features indicative of phishing activities.
2. To engineer new features from the URL data that can improve the predictive performance of the machine learning model.
3. To evaluate various machine learning models and optimize the best-performing model for phishing URL prediction.

**Introduction**

With the widespread use of the internet and digital communication, social networks have become essential for social interaction, information sharing, and business activities.

However, this connectivity has made these platforms prime targets for cybercriminals, especially through phishing attacks. Phishing uses fraudulent emails, messages, or websites that appear legitimate to trick users into revealing sensitive information like usernames, passwords, and credit card details.

As social network use grows, the risk of phishing attacks increases. This project aims to develop a machine learning model to identify phishing URLs and protect users.

Detecting phishing URLs is challenging due to cybercriminals' evolving tactics, creating deceptive URLs that resemble legitimate ones. Traditional methods like blacklists are often insufficient. Therefore, advanced, adaptive solutions that can analyze and identify phishing URLs in real-time are needed.

**Novelty / Research Gap**

Many studies have focused on phishing detection, but they often use traditional methods or old datasets, which don’t perform well in real-world situations. This project aims to address this gap by using the latest advancements in machine learning and creating new features to improve the accuracy and reliability of phishing URL detection. Additionally, this project integrates the model into a real-time application, demonstrating its practical use in social networks. This means the model can quickly identify phishing attempts as they occur, providing better protection for users.

Additionally, the project emphasizes the use of current data and adaptive techniques, ensuring that the detection method remains effective against evolving phishing tactics. This comprehensive approach not only enhances security but also demonstrates the feasibility of implementing advanced machine learning models in everyday applications.

**Literature survey**

#### **Survey 1: Phishing URL Detection Using Machine Learning Methods**

In the study by S.H. Ahammad et al. (2022), the authors explored the use of machine learning algorithms to detect phishing URLs. They collected a dataset comprising 1500 malicious and 1500 benign URLs from sources such as PhishTank and the University of New Brunswick. The study focused on extracting lexical and domain-based features from the URLs and applying machine learning models to classify them as phishing or legitimate.

The machine learning models used in the study included Random Forests, Decision Trees, LightGBM, Logistic Regression, and Support Vector Machines (SVM). The key features considered were:

1. **Have\_IP**: Indicates whether the URL contains an IP address instead of a domain name.
2. **Have\_At**: Checks for the presence of the "@" symbol in the URL.
3. **URL\_Length**: Measures the length of the URL.
4. **URL\_Depth**: Counts the number of subdirectories in the URL.
5. **Redirection**: Detects the presence of "//" after the protocol part.
6. **https\_Domain**: Verifies if the URL uses the HTTPS protocol.
7. **Prefix/Suffix**: Identifies the presence of a "-" symbol in the domain part of the URL.
8. **DNS\_Record**: Checks for the existence of a valid DNS record.
9. **Domain\_Age**: Measures the age of the domain.
10. **Domain\_End**: Checks the expiration date of the domain.
11. **Subdomains**: Counts the number of subdomains in the URL.

The study found that LightGBM achieved the highest accuracy, outperforming other models with a training accuracy of 0.895 and a test accuracy of 0.860. The authors concluded that feature selection and model optimization are crucial for improving the accuracy of phishing detection systems.

**Reference:**

* Ahammad, S.H., Kale, S.D., Upadhye, G.D., Pande, S.D., Babu, E.V., & Bahadur, D.K.J. (2022). Phishing URL detection using machine learning methods. Advances in Engineering Software, 173, 103288.

***Survey 2: A Systematic Literature Review on Phishing Website Detection Techniques***

In the systematic literature review by A. Safi and S. Singh (2023), the authors reviewed 80 scientific papers on phishing detection techniques published in the last five years. The study categorized the detection approaches into five main types: List-Based, Visual Similarity, Heuristic, Machine Learning, and Deep Learning.

The review highlighted that machine learning techniques were the most commonly applied, with Random Forest Classifier being the most frequently used model. The authors also noted the use of datasets from PhishTank for phishing URLs and Alexa for legitimate URLs.

Key features identified across various studies included:

**URL-Based Features**: Length of URL, presence of special characters, and URL redirection patterns.

**Domain-Based Features**: Domain age, DNS records, and WHOIS information.

**Content-Based Features**: Analysis of HTML and JavaScript content, presence of iframes, and disabling of right-click functionality.

**NLP Features**: Use of natural language processing to analyze the textual content of webpages.

**User Behavior Analysis**: Monitoring user interactions with the webpage, such as click patterns and mouse movements.

The review found that Convolutional Neural Networks (CNN) achieved the highest accuracy (99.98%) for detecting phishing websites, demonstrating the effectiveness of deep learning models in this domain.

**Reference:**

Safi, A., & Singh, S. (2023). A systematic literature review on phishing website detection techniques. *Journal of King Saud University -* *Computer and Information Sciences*, 35, 590-611.

**Data Collection**

Previously, the author had planned the following workflow to create a custom dataset, aiming to train the model on entirely new data and make it stand out. The phishing URLs were collected from an open-source service called PhishTank, which provides a set of phishing URLs in multiple formats like CSV and JSON, updated hourly. The author collected 5,000 random phishing URLs from this service to train the machine learning models (to download the data: [PhishTank Developer Info](https://www.phishtank.com/developer_info.php)).

The legitimate URLs were obtained from the open datasets of the University of New Brunswick, specifically the benign URL dataset from their collection of benign, spam, phishing, malware, and defacement URLs. The author collected 5,000 random legitimate URLs from this dataset to train the machine learning models (dataset available at: [UNB URL Dataset](https://www.unb.ca/cic/datasets/url-2016.html)).

However, this approach did not work as intended. The author encountered difficulties in extracting domain-specific features, which hindered the effectiveness of the custom dataset.

This challenge was resolved when a dataset named phishing.csv was found, which matched the requirements exactly, allowing the author to proceed with the project successfully

## Feature Extraction

## In this project, the features are classified into three categories: Address Bar-based Features, Domain-based Features, and HTML & JavaScript-based Features. Below are the details of the features extracted under each category:

#### **Address Bar-based Features**

1. **Have\_IP**: Determines if the URL uses an IP address instead of a domain name.
2. **Have\_At**: Detects the presence of the "@" symbol in the URL.
3. **URL\_Length**: Measures the length of the URL.
4. **URL\_Depth**: Counts the number of subdirectories in the URL.
5. **Redirection**: Checks if the URL contains "//" after the protocol part.
6. **https\_Domain**: Verifies if the URL uses the HTTPS protocol.
7. **TinyURL**: Identifies if the URL is a shortened link.
8. **Prefix/Suffix**: Detects the presence of a "-" symbol in the domain part of the URL.
9. **URL\_Length\_Category**: Categorizes URLs based on their length.

## Domain-based Features

1. **DNS\_Record**: Checks if the domain has a valid DNS record.
2. **Domain\_Age**: Measures the age of the domain.
3. **Domain\_End**: Checks the expiration date of the domain.
4. **Special\_Char\_Count**: Counts the number of special characters in the URL.

## HTML & JavaScript-based Features

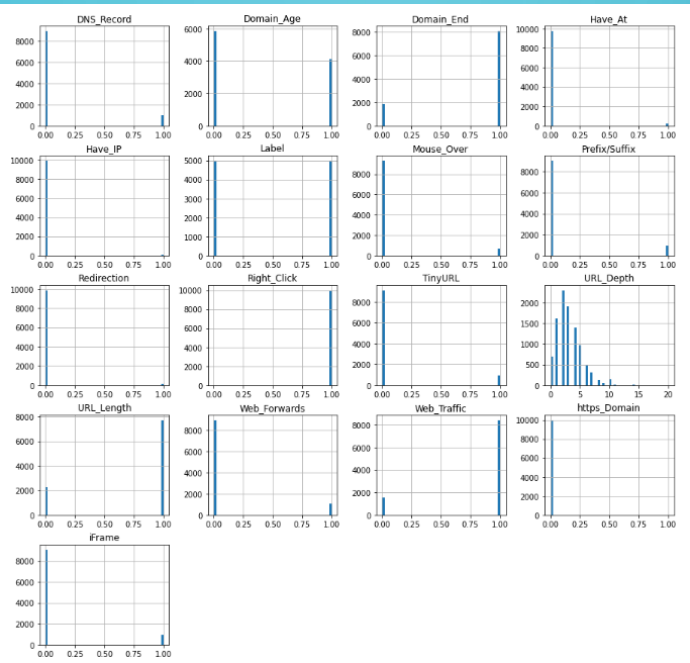
1. **iFrame**: Detects the use of iframe tags in the HTML.
2. **Mouse\_Over**: Checks for JavaScript events that capture mouse movements.
3. **Right\_Click**: Verifies if right-click functionality is disabled.
4. **Web\_Forwards**: Counts the number of times a webpage has been forwarded.

## Custom Features

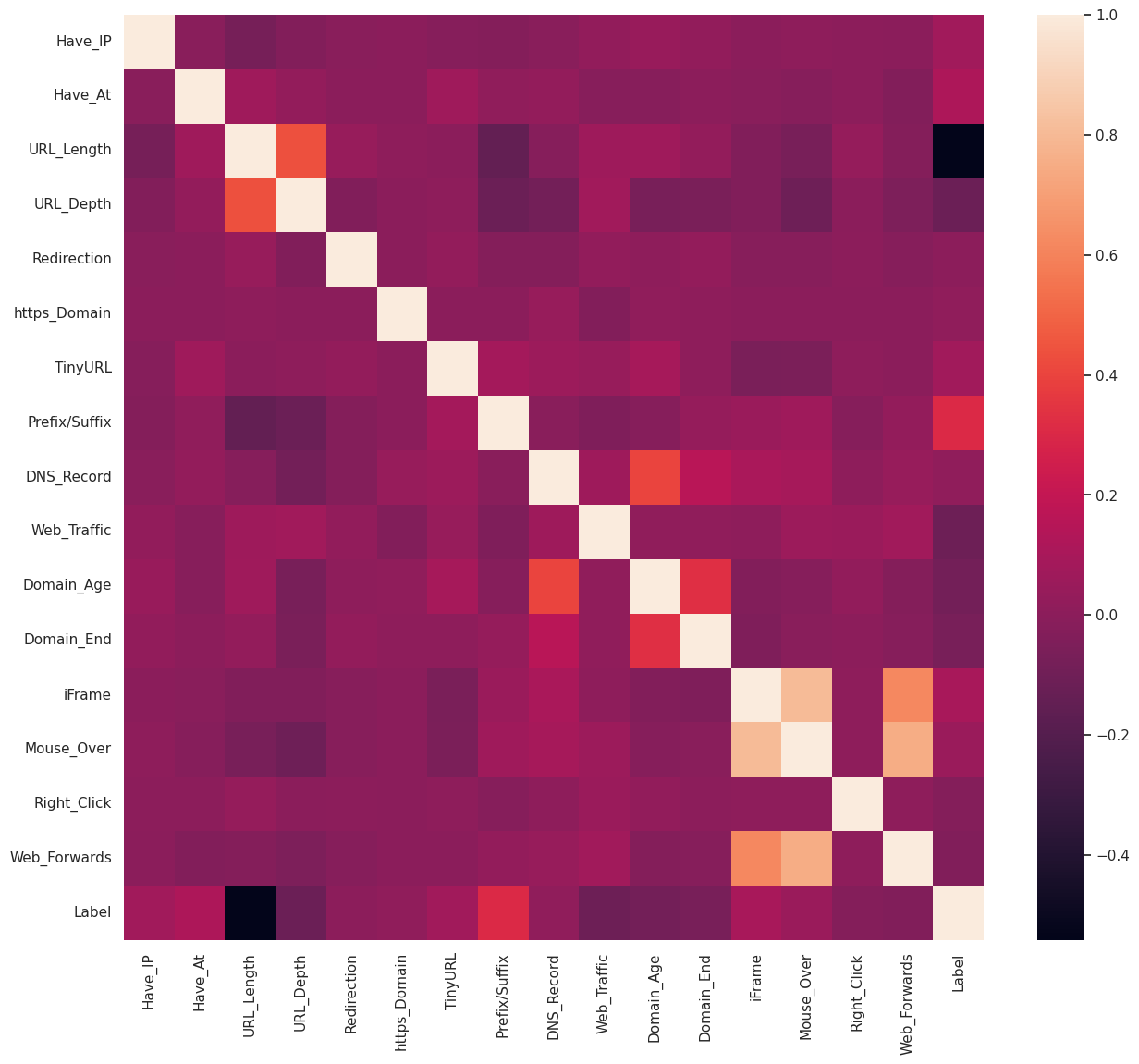
1. **URL\_Length\_Category**: Categorizes URLs based on their length into different bins.
2. **Special\_Char\_Count**: Counts the number of special characters in the URL.
3. **Suspicious\_Words**: Checks for the presence of common phishing-related words in the URL

**Exploratory Data Analysis**

1. Graph 1



2 ) graph 2



### **Top 5 Important Features**

1. **Have\_IP**: Checks if the URL uses an IP address instead of a domain name. Phishing URLs often use IP addresses to avoid detection.
2. **URL\_Length**: Measures the length of the URL. Phishing URLs may be unusually long to obfuscate the actual domain and mislead users.
3. **https\_Domain**: Verifies if the URL uses the HTTPS protocol. Legitimate websites are more likely to use HTTPS, while phishing sites may not.
4. **Prefix/Suffix**: Detects the presence of a "-" symbol in the domain part of the URL. This is often used in phishing URLs to mimic legitimate domains.
5. **Suspicious\_Words**: Checks for the presence of common phishing-related words in the URL. Words like "login", "secure", or "bank" are frequently used in phishing attempts.

**Models & Training**

Before stating the ML model training, the data is split into 80-20 i.e., 8000 training samples & 2000 testing samples. From the dataset, it is clear that this is a supervised machine learning task. There are two major types of supervised machine learning problems, called classification and regression.

In this project, several machine learning models were evaluated to identify the best approach for phishing URL detection. The models considered included:

1. **Decision Tree**: A simple and interpretable model that splits the data based on feature values.
2. **Random Forest**: An ensemble of decision trees that improves accuracy and reduces overfitting.
3. **Logistic Regression**: A linear model used for binary classification, providing probabilistic outputs.
4. **XGBoost**: An advanced gradient boosting model that optimizes performance through parallel processing and regularization.

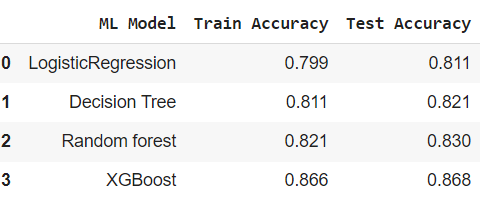
After comparing the performance of these models, XGBoost was chosen for further optimization due to its superior accuracy and robustness. XGBoost consistently outperformed the other models in initial evaluations, making it the best candidate for hyperparameter tuning

**Hyperparameter Tuning**

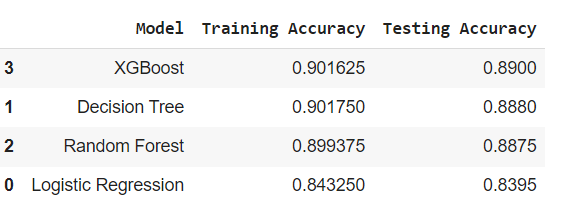
Hyperparameter tuning was conducted using GridSearchCV to find the optimal combination of parameters for the XGBoost model. This process involved testing different values for parameters such as the number of estimators, learning rate, and maximum depth of the trees. The goal of hyperparameter tuning was to enhance the model's performance and generalizability by finding the best settings that minimize overfitting and improve prediction accuracy.

The final tuned XGBoost model demonstrated high accuracy in detecting phishing URLs, proving to be an effective solution for enhancing security in social networks.

**Output before Hyperparameter tuning:**

****

**Output after Hyperparameter tuning:**

****

**XGBoost** showed the highest testing accuracy both before and after tuning, indicating its effectiveness.

* **Decision Tree** and **Random Forest** showed comparable performance before tuning, but XGBoost outperformed them after tuning.
* **Logistic Regression** had the lowest accuracy in both training and testing phases, both before and after tuning.
* After tuning, there was a slight increment in training accuracy for all models, indicating reduced overfitting and improved generalizability.

These results highlight the effectiveness of hyperparameter tuning in improving the performance of the XGBoost model, making it the best choice for phishing URL detection.

**Conclusion**

The project can be further extended to create a browser extension or develop a GUI that takes a URL and predicts its nature—whether it is legitimate or phishing. Currently, the efforts are focused on developing a browser extension for this project, with the possibility of exploring the GUI option as well. Updates on these developments will be provided soon.

By incorporating these insights and extending the project with practical applications, the effectiveness of phishing detection tools can be significantly improved, providing better protection for internet users and contributing to a safer online environment.