# **Phishing detection using machine learning**

|  |  |
| --- | --- |
| **Written By** | **Om Singh,Shaileja Patil** |
| **Document Version** | **0.1** |
| **Last Revised Date** | **28 January 2024** |

**Document Control**

### Record:

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Reviewer** | **Comments** |
| 0.1 | 23 – January -  2024 | Shaileja Patil | Data Collection ,Data Cleaning ,Data Transformation |
| 0.1 | 24 – January -  2024 | Om Singh,  Shaileja Patil | Data Transformation ,Feature Selection |
| 0.1 | 25 – January -  2024 | Om Singh | HyperPrameter tunning(Using HyperOPT,TPOTClassifier,Optuna,  GridSearchCV),ModelCreation |
| 0.1 | 26 – January -  2024 | Om Singh,  Shaileja Patil | Hyperparameter Tunning Modify the parameter to Increase the accuracy,ModelCreation,ModelTesting |
| 0.1 | 27 – January -  2024 | Om Singh | Creating Fronent Appliation using Streamlit and deployment |
| 0.1 | 6 February 2024 | Om Singh, Shaileja Patil | Updateing final model ,Creating LLD,HLD,Readme and others. |

### Reviews:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Review**  **Date** | **Reviewed By** | **Approved By** | **Comments** |
|  |  |  |  |  |

### Approval Status:

Contents

1. Introduction

2. Scope

3. Architecture

3.1 Overview

3.2 System Components

3.2.1 Data Exploration

3.2.2 Data Cleaning

3.2.3 Feature Engineering

3.3 Model Building

3.4. Model Testing

1. Developing Frontend Application
   1. Develop a Frontend applicationusing Streamlit
2. Deployment  
    The Project is Deployed using Render
3. Conclusion

6.1 Summary of Findings

6.2 Lessons Learned

6.3 Recommendations for Future Work

# Introduction

Phishing attacks continue to be a persistent and evolving threat in the digital landscape, posing significant risks to individuals, organizations, and their sensitive information. These attacks often involve the use of deceptive URLs that mimic legitimate websites, tricking users into divulging confidential data such as usernames, passwords, and financial details. As the sophistication of phishing attacks increases, there is a growing need for robust and proactive measures to detect and mitigate these threats.

The "Phishing URL Detection Project" aims to develop an advanced system that can effectively identify and flag malicious URLs associated with phishing activities. Leveraging cutting-edge technologies and machine learning algorithms, this project seeks to provide a proactive defense against phishing attempts, safeguarding users from falling victim to fraudulent schemes.

# Scope :

The scope of the Phishing URL Detection Project encompasses various key aspects, including:

URL Analysis: The project will involve in-depth analysis of URLs to identify patterns, characteristics, and anomalies associated with phishing attempts. This analysis will encompass both structural components and content-based features of URLs.

Machine Learning Models: Advanced machine learning models will be employed to train the system to recognize patterns indicative of phishing URLs. Supervised learning techniques will be utilized to classify URLs into either legitimate or malicious categories based on historical data and features extracted during the analysis.

Feature Extraction: The project will explore various features extracted from URLs, such as domain reputation, lexical features, and behavioral patterns, to enhance the accuracy and efficiency of the detection models.

Real-time Detection: The system will be designed to operate in real-time, providing instantaneous detection and response to phishing threats. This will involve the integration of the detection model into web browsers, email clients, or other relevant applications.

Scalability: The project will consider scalability to ensure the system can handle a large volume of URLs across different platforms, making it suitable for deployment in diverse environments.

User Education: While the primary focus is on technical solutions, the project will also explore opportunities for user education and awareness to complement the automated detection system. This may include providing users with information on recognizing and reporting phishing attempts.

# Architecture Description

3.1. Overview

The architecture of the project is designed to encompass a series of

well-defined stages, ensuring a robust approach to phishing detection using machine learning. The primary components include data exploration, data cleaning, feature engineering, model building, and model testing.

3.2. System Compounds

3.2.1 Data Exploration

In the Data Exploration phase, the objective is to gain a deep understanding of the dataset. Exploratory Data Analysis (EDA) techniques are employed, including statistical measures, data visualization, and identifying potential outliers. Key features selected during this phase may include time-based trends, frequency distributions, and data distribution visualizations.

3.2.2 Data Cleaning

Data Cleaning is a critical step to enhance the quality and usability of the dataset. Techniques such as dropping columns based on correlation, identifying redundant columns, and

handling columns with zero variance are applied. Selected features from this phase include:

- Highly Correlated Columns:"click\_count," "hover\_time"

- Non-Redundant Columns:"domain\_age," "registration\_length"

- Non-Zero Variance Columns:"phishing\_attempts","webpage\_length"

3.2.2 Feature Engineering

Feature Engineering is a crucial aspect to derive meaningful information from the data. The following subcomponents are employed:

3.3. Model Building

In the Model Building phase, classical machine learning algorithms are applied. Hyperparameter tuning is performed using Hyperopt, Optuna, and TPOT to optimize the model's performance. Selected features from the previous phases are used to train the model, ensuring a holistic representation of the dataset.

3.4. Model Testing

Model Testing involves evaluating the model's performance on different datasets. Precision, accuracy, recall, and F2 score are utilized as evaluation metrics. The model is tested on:

- Train Data

- Test Data

- Validation Data(dataset\_small.csv)

# Develop a Frontend Application

4.1 Developing a Frontend Application using Streamlit

The frontend application serves as an interface for users to interact with the phishing detection model. Streamlit, a user-friendly Python library, is employed to create a dynamic and intuitive user interface. The application allows users to input URLs or data, triggering the model's predictions. Key features of the frontend application include:

- User-Friendly Input: A simple and intuitive input form that enables users to submit URLs or relevant data for analysis.

- Prediction Display: Clear presentation of the model's predictions, indicating whether a given input is classified as a potential phishing attempt

# Deployment

The deployment phase involves making the phishing detection model

and the frontend application accessible to end-users. This is

Typically done through Render.

# Conclusion

6.1 Summary Of findings

In summary, the project has successfully addressed the challenge

of phishing detection through a systematic approach. Key findings include:

- The effectiveness of various features in distinguishing between legitimate and malicious domains.

- The model's performance metrics, including precision, accuracy, recall, and F2 score, demonstrating its capability to accurately identify phishing attempts.

6.2 Leason Learned

Throughout the project, several valuable lessons have been learned, including:

- The importance of feature engineering in enhancing the model's predictive capabilities.

- The significance of continuous model evaluation and refinement

to adapt to evolving phishing tactics.

- The usability and impact of a user-friendly frontend application

for real-world applications.

6.3 Recomandation for Future work

While the current project provides a robust solution for phishing detection, there are opportunities for future enhancements,such as:

- Continuous Model Updates: Implementing mechanisms for real-time updates to the model based on new data and emerging trends.

- Advanced Feature Engineering: Exploring more advanced feature engineering techniques to capture nuanced patterns in phishing

attempts.

- User Feedback Integration: Incorporating user feedback from the frontend application to further improve the model's accuracy.

This comprehensive conclusion highlights the achievements, lessons earned, and potential avenues for future research and development.

# Thank You