# **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. |

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

Contents

1. Introduction

1.1 Background

1.2 Problem Statement

1.3 Objectives

2. Scope

2.1 In-scope

2.2 Out-of-scope

2.3 Assumptions

3. Architecture

3.1 Overview

3.2 System Components

3.2.1 Data Exploration

3.2.2 Data Cleaning

3.2.3 Feature Engineering

-3.2.3.1 URL-Based Features

- 3.2.3.2 Domain-Based Features

- 3.2.3.3 Page-Based Features

- 3.2.3.4 Content-Based Features

3.3 Model Building

3.3.1 Performing hyperparameter tunning

3.4. Model Testing

1. Developing Frontend Application
   1. Develop a Frontend application using Streamlit
2. Deployment

5.1 Render Deployment

1. Conclusion

6.1 Summary of Findings

6.2 Lessons Learned

6.3 Recommendations for Future Work

# Introduction

## What is the Background for the Project?

In today's digital age, phishing has emerged as a

prevalent and sophisticated form of fraud. Cyber

attackers employ deceptive tactics, often impersonating reputable entities, to trick individuals into divulging sensitive information such as login credentials and

personal data. This project addresses the pressing

need for robust cybersecurity measures by leveraging

machine learning techniques to predict the authenticity

of domains.

## Problem Statement

The proliferation of phishing attacks poses a significant threat to online security.The challenge is to distinguish between legitimate and fake domains accurately.Legitimate domains are crucial for communication and business, while fake domains are often employed by malicious actors for phishing schemes.Effectively tackling this problem is essential for enhancing overall cybersecurity and safeguarding users from falling victim to phishing attacks

## Objectives

The primary objective of this project is to develop a predictive model that goes beyond traditional security measures. The aim is to create a sophisticated system that can accurately classify domains as either real or fake. By employing classical machine learning techniques, the model seeks to contribute to a proactive defense against phishing attacks.

# Scope

2.1. In Scope

The project includes the following key elements:

- Data Collection:

Gathering relevant data sources related to phishing

attempts.

- Data Cleaning:

Preprocessing the collected data to ensure its quality

and reliability.

- Feature Selection:

Identifying and extracting meaningful features for

model training.

- Model Development:

Utilizing classical machine learning techniques for

creating a predictive model.

- Model Testing:

Evaluating the model's performance using appropriate

metrics.

- Frontend Application:

Developing a frontend application using Streamlit for

user interaction.

- Deployment:

Deploying the trained model and the frontend application.

2.2. Out of Scope

The following aspects are considered out of the project's current scope:

- Implementation of Specific Security Measures: While the project focuses on phishing detection, the actual implementation of specific security measures based on the model's predictions is beyond the scope.

- Real-time Data Updates: Continuous updates to the training data in real-time are not within the project's scope. The model will be trained on a fixed dataset, and periodic retraining is recommended for improved accuracy.

- Real-time Data Updates: Continuous updates to the training data in real-time are not within the project's scope. The model will be trained on a fixed dataset, and periodic retraining is recommended for improved accuracy.

2.3. Asumption

The successful execution of the project relies on the following assumptions:

- Representative Dataset: It is assumed that the provided dataset is representative of real-world scenarios concerning phishing attempts.

- Applicability of Classical ML Techniques: The project assumes that classical machine learning techniques are suitable for the given task of phishing detection.

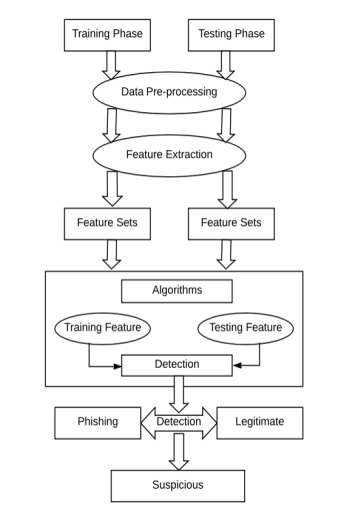
- Static Model Training: The project operates under the assumption that the model training process is static, and periodic retraining will be conducted to adapt to evolving phishing tactics.

- User Interaction: The frontend application assumes a certain level of user interaction for input and output. The success of the user interface is contingent on user understanding and cooperation.

# 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

3.2.2 Feature Engineering

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

3.2.2.1 Url Based Feature

URL-Based Features are derived to capture patterns and characteristics from the URLs in the dataset. Selected

features may include:

- URL Length: "url\_length"

- Special Characters: "special\_char\_count"

- Subdomain Count: "subdomain\_count"

3.2.2.2 Domain Based Feature

Domain-Based Features involve extracting information specific

to domains. Features such as:

- Domain Age: "domain\_age"

- Registration Length: "registration\_length"

3.2.2.3 Page Based Feature

Page-Based Features focus on characteristics extracted from

the webpage content. Selected features may include:

- Hyperlink Count: "hyperlink\_count"

- JavaScript Presence: "javascript\_presence"

- Text-to-HTML Ratio: "text\_html\_ratio"

3.2.2.4 Content Based Feature

Content-Based Features involve analyzing the overall content

to identify patterns and potential phishing signals. Selected features may include:

- Keyword Frequency: "phishing\_keyword\_count"

- External Link Presence: "external\_link\_presence"

3.2.2.5 Other Feature

- File-Based Features: Characteristics related to file attachments, assessing file type, size, and presence.

- Parametric-Based Features: Attributes derived from

parameters in URLs or web requests.

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.

3.3.1 Hyperparameter Tunning:

We tried various hyperparameter tnning method (GridSearcg ,hyperopt,tpot and even optuna on various model like RndomForestClassifier,Xgbboost,KNN ,and even Kmeans

After training are model on various parameter we conclude that the Knn Model is best suited for our motive of classification we achieved a Test Accuracy of almost 95% with Knn,

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 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 phishingdetection, 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.

6.3 Problem Faced During training :

While training we got to know that the dataset contains many dupicates The origina size of dataset was 88457 which droped to almost 28000 when we droped the duplicates.

# Thank You