**Architecture**

**“PHISHING DOMAIN Detection”**

|  |  |
| --- | --- |
| Written by | Sahil Sharma |
| Document version | 1.0 |
| Last Revision Date | 10/06/2024 |

**Document Control**

***Contents*   *Page no.***

1. Introduction

[1.1. What is Low Level Design Document? 3](#_Toc8145)

[1.2. Scope 3](#_Toc8146)

[2. Architecture 4](#_Toc8147)

[3. Architecture Description](#_Toc8148)

[3.1. Data Description 5](#_Toc8149)

[3.2. Import Data 5](#_Toc8150)

[3.3. Data Cleaning 5](#_Toc8151)

[3.4. Exploratory Data Analysis 5](#_Toc8152)

[3.5. Data Preprocessing 5](#_Toc8153)

[3.6. Model Building 5](#_Toc8154)

[3.7. Model Dump 5](#_Toc8155)

[3.8. Data From User 5](#_Toc8156)

[3.9. Data Validation 5](#_Toc8157)

[3.10. Model Call for Specific Input 6](#_Toc8158)

1. **Introduction**

The ever-growing digital landscape brings immense convenience, but also exposes us to evolving cyber threats like phishing. Phishing attacks attempt to deceive users into entering sensitive information on fraudulent websites that appear legitimate. These attacks can steal login credentials, financial data, or even personal information.

This project aims to develop a system capable of predicting phishing domains. By proactively identifying suspicious websites, we can create a safer online environment for everyone.

## 1.1 Why this Architecture Design Document?

This architecture design document serves as the technical blueprint for our phishing domain prediction system. It outlines the system's structure, detailing its various components, functionalities, and how they interact. Here's why this document is crucial:

**Informed Decision-Making**: By outlining the architectural choices, along with their advantages and limitations, this document empowers stakeholders to make informed decisions regarding the system's development and future enhancements.

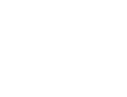
**Maintainability and Scalability**: A well-defined architecture lays the foundation for a maintainable and scalable system. This document serves as a reference for future developers who need to understand, modify, or expand the system's capabilities.

**Reduced Risk of Errors:** A documented architecture helps identify and mitigate potential design flaws early in the development process. This reduces the risk of errors that could lead to costly rework or security vulnerabilities.

# 1 Architecture

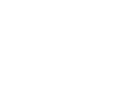


Start



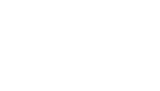
Data

Collection



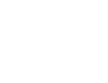
Data

Validation



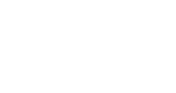
Data

Cleaning



Get Best

Model



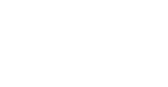
Data

Pre-processing



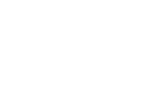
Export

Data



Data

Insertion



Model

Saving

Creation

OUTPUT



End



# 2 Architecture Description

## 3.1 Data Description

Phishing domain data is crucial for training and evaluating machine learning models designed to identify phishing attempts. This data describes the characteristics of websites used for phishing scams and can be categorized into three main aspects.

## 3.2 Import Data

Data Import from Database - The data in a stored database is imported as a CSV file to be used for Data Pre-processing and Model Training.

## 

## 3.3 Data Cleaning

There are no null values in the data and all the variables are categorical, some of the observations meaningless and they are converted into meaningful observation. **Example:** “?” is converted into letter “m” (Missing)

## 3.4 Exploratory Data Analysis

Every independent variable in the dataset is displayed by a multiple bar plot as it relates to the dependent variable's classes of Phishing and Legitimate website as part of the EDA process.

## 3.5 Data Pre-processing

Data pre-processing steps are converting categorical variables into numerical variables using label encoding method and train and test split of the data etc.

## 3.6 Model Building

Following data pre-processing, separate the data into train and test sets (using simple random sampling), and use several machine learning algorithms for classification. (Random Forest, Decision Tree, Logistic regression, Naïve Bayes, KNN)

## 3.7 Model Dump

I developed a model and used the pickle module to dump the model in a pickle file format after comparing all accuracy levels and determining the optimal model for the dataset.

## 3.8 Data from User

Here with the aid of the UI interface, the user must input the names of all the features in the proper order and submit it to the model. The model will be fed the data and determine whether or not the feature set depicting a mushroom is edible.

## 3.9 Data Validation

Here Data Validation will be done, given by the user.

## 3.10 Model Call for specific input

A NumPy array will be created based on the User Input and fed to our model after being sent to the backend in variable format. After the pickle file has been loaded, the model will determine whether the input is edible or not and will communicate the conclusion to our html page.