**Low Level Design**

**“PHISHING DOMAIN DETECTION”**

|  |  |
| --- | --- |
| Written by | **SAHIL SHARMA** |
| Document version | 0.3 |
| Last Revision Date | 010 - 06 -2024 |

**Document Control**

|  |  |  |
| --- | --- | --- |
| **Date** | **Version** | **Author** |
| 010- 06 -2024 | **0.3** | SAHIL SHARMA |

***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)

[4. Technology Stack 7](#_Toc8161)

**1. Introduction**

## 1.1 What is Low Level Design Document?

Giving the internal logical design of the actual programme code is the aim of a low-level design document (LLD). Based on the high-level design, the low-level design is produced. LLD explains class diagrams that show the relationships and methods between classes and programme specifications. In order for the programmer to create the programme directly from the document, it describes the modules.

## 1.2 Scope

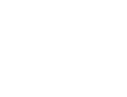
Low-level design (LLD) is a component-level design method that incorporates constant iteration and improvement. Data structures, necessary software architecture, source code, and finally performance algorithms can all be designed using this method. Overall, during requirement analysis, the data organisation may be created, and then refined, during data design work. Each component is precisely specified after the build.

# 

# Architecture

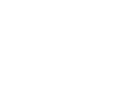


Start



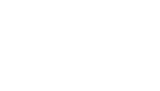
Data

Collection



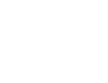
Data

Validation



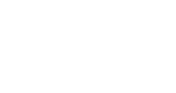
Data

Cleaning



Get Best

Model



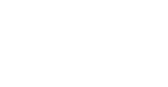
Data

Pre-processing



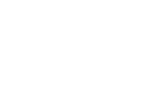
Export

Data



Data

Insertion



Model

Saving

Creation

OUTPUT



End



# Architecture Description

## 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.

## Import Data

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

## 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)

## 

## 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 Websites as part of the EDA process.

## 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.

## Model Building

After Data pre-processing split the data train and test (Simple Random Sampling) and implemented different Classification Machine Learning Algorithm. Checking which model gives us the best accuracy. (Random Forest, SVM, Decision Tree, Logistic regression, Naïve Bayes, KNN)

## 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.

## Data from User

Here, using the UI interface, the user must input all the feature values in the proper order before sending it to the model. The model, which will determine whether the feature set accurately represents the desired features, will be fed the data.

## Data Validation

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

## Model Call for specific input

Based on the user's input, the data will be handled in the backend with a variable format before being converted into a NumPy, array and provided to an ML model. The model will check the inputs against the required criteria after loading the pickle file by sending the output to our html page.

# 4.Technology Stack

|  |  |
| --- | --- |
| **Back End** | Pandas, NumPy, Sci-kit learn, etc |
| **Deployment** | Local host |