Low Level Design (LLD)

Phishing Domain Detection

April 1, 2022

Abstract

Phishing is a type of fraud in which an attacker impersonates a reputable company or person in order to get sensitive information such as login credentials or account information via email or other communication channels. Phishing is popular among attackers because it is easier to persuade someone to click a malicious link that appears to be authentic than it is to break through a computer's protection measures.

Hence creating a phishing detection solution will help web users to stay away from phishing attempts.

Revision number: 1.0

Last date of revision: April 1, 2022

Contents

Chapter 1

Introduction

1.1 Purpose of LLD

Return to table of contents

This document is aimed to provide a detailed description of phishing detection solution. It will explain the aim and features of the solution, interfaces of the solution, what the solution will do, and constraints under which it must operate. This document is for both stakeholders and developers of the solution.

The goal of the project is to identify malicious websites.

1.2 Scope

Return to table of contents

The solution will be a web application. It will take a website URL or a csv/xlsx file with URLs as input and return predictions (a single output or a csv file) telling whether a website/websites is/are malicious or not.

1.3 Constraints

Return to table of contents

The dataset contains only 88,647 observations, which may seem to be a good set of observations. But in the era of big data it is not that impressive in terms of data size.

Chapter 2

Technical specifications

2.1 Dataset

Return to table of contents

The data contains 88,647 observations of websites. Specifically 58,000 legitimate and 30,647 phishing website instances with 111 features. The data comes as a csv file. The data is hosted here.

2.2 Phishing detection

Return to table of contents

1. The system displays input required.
2. User provides the input value (a website URL or a csv/xlsx file with URLs).
3. The system predicts whether a website/websites is/are malicious or not and displays/outputs prediction/predictions.

2.3 Logging

Return to table of contents

The logging is applied in every process, but logging results will not be accessible for end users. Moreover logging is made accessible via console and a file for developers. Logging is mandatory as it allows to debug issues more easily.

2.4 Database

Return to table of contents

The database is hosted on Astra DB. The system does not store any request into the database except via logging.

2.5 Deployment

Return to table of contents

HEROKU

2.6 Technology stack

Return to table of contents

Front end	HTML/CSS
Back end	Flask
Database	Cassandra
Deployment	Heroku
CI-CD	CircleCi

Chapter 3

Proposed solution

3.1 Description

Return to table of contents

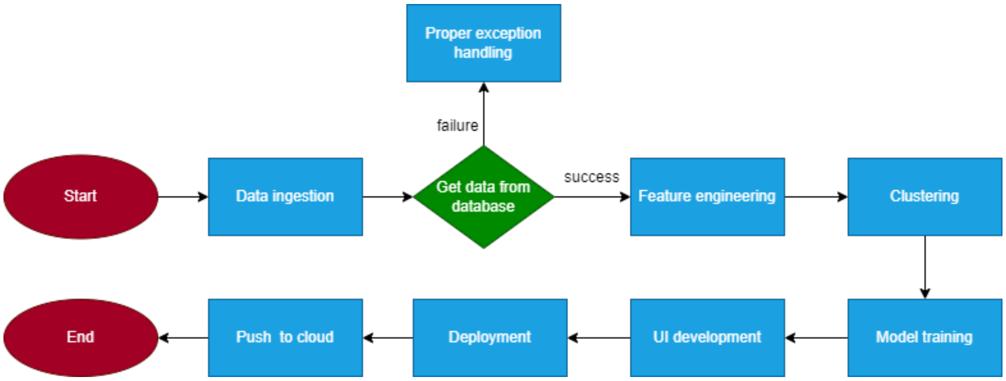
As it is possible that for some subsets of the data an algo will dominate other algos in terms of prediction accuracy (ROC-AUC), first of all KMeans is applied to detect clusters whithin the data. Solution building includes finding the best model for each cluster. We use Support Vector Machine, XGBoost, Naive Bayes, Logistic Regression, Random Forest, and LightGBM algorithms. For each cluster, the algorithms use the best parameters derived from GridSearch/BayesSearch. We calculate the ROC-AUC scores for the models and select the model with the best score for each cluster.

Chapter 4

Model training/validation workflow

4.1 Diagram

Return to table of contents

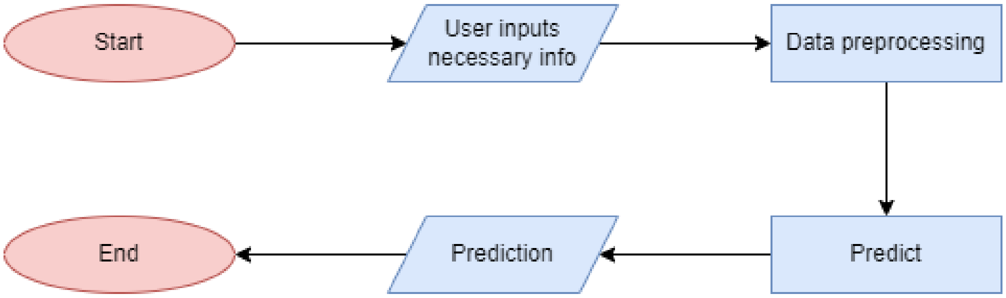


Chapter 5

User I/O workflow

5.1 Diagram

Return to table of contents



Chapter 6

Test cases

6.1 Description

Return to table of contents

Testing has not been conducted.