High Level Design (LLD)

Phishing Domain Detection

Revision Number: 1
Last date of revision: 21/08/2022

Saurabh Naik

Document Version Control

Date Issued	Version	Description	Author
21th August 2022	1.1	First Draft	Saurabh Naik

Table of Contents

Document Version Control 2
Abstract 4
1.Introduction 5
1.1 Why this High-Level Design Document? 5
1.2 Scope 6
1.3 Constraints 6
1.4 Risks 6
1.5 Out of Scope 6
2.Technical specifications 7
2.1 Dataset 7
2.2 Dataset overview 7
2.3 Logging 8
2.4 Database 8
3. Deployment 9
4.Technology stack 10
5. Proposed Solution 11
6. Proposed Architecture 12
7. Model training/validation workflow 13
8.User I/O workflow 14
9. Error Handling 15
10. Model performances 16
11. Key performance indicators (KPI) 17
12. Conclusion 18
13. References 19

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 The mail goal of this end to end application is to predict whether the domains are real or malicious using various combination of machine learning algorithms by dividing the dataset into various clusters and applying a suitable algorithm for that particular cluster.

1.Introduction

1.1 Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding, and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

- Present all of the design aspects and define them in detail
- Describe the user interface being implemented
- Describe the hardware and software interfaces
- Describe the performance requirements
- Include design features and the architecture of the project
- List and describe the non-functional attributes like:
    - Security
    - Reliability
    - Maintainability
    - Portability
    - Reusability
    - Application compatibility
    - Resource utilization
    - Serviceability

1.2 Scope

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system. This software system will be a Web application this system will be designed to detect malicious websites.

1.3 Constraints

We will only be detecting malicious websites.

1.4 Risks

Document specific risks that have been identified or that should be considered.

1.5 Out of Scope

Delineate specific activities, capabilities, and items that are out of scope for the project.

2.Technical specifications

2.1 Dataset

Data Set Name	Finalized	Source
Dataset_full.csv	Yes	Phishing Websites Dataset - Mendeley Data

2.2 Dataset overview

- These data consist of a collection of legitimate as well as phishing website instances. Each website is represented by the set of features which denote whether website is legitimate or not. Data can serve as an input for machine learning process. In this repository the two variants of the Phishing Dataset are presented. Full variant - dataset_full.csv Short description of the full variant dataset: Total number of instances: 88,647 Number of legitimate website instances (labelled as 0): 58,000 Number of phishing website instances (labelled as 1): 30,647 Total number of features: 111

2.3 Logging

We should be able to log every activity done by the incidents.

- The System identifies at what step logging required
- The System should be able to log each and every system flow.
- Developers can choose logging methods. You can choose database logging/ File logging as well.
- System should not be hung even after using so many loggings. Logging just because we can easily debug issues so logging is mandatory to do.

2.4 Database

System needs to store every request into the database and we need to store it in such a way that it is easy to retrain the model as well.

1. The User chooses the activity dataset.
2. The User gives required information.
3. The system stores each and every data given by the user or received on request to the database. Database chosen in this case is Cassandra Database.

3. Deployment

1. Heroku

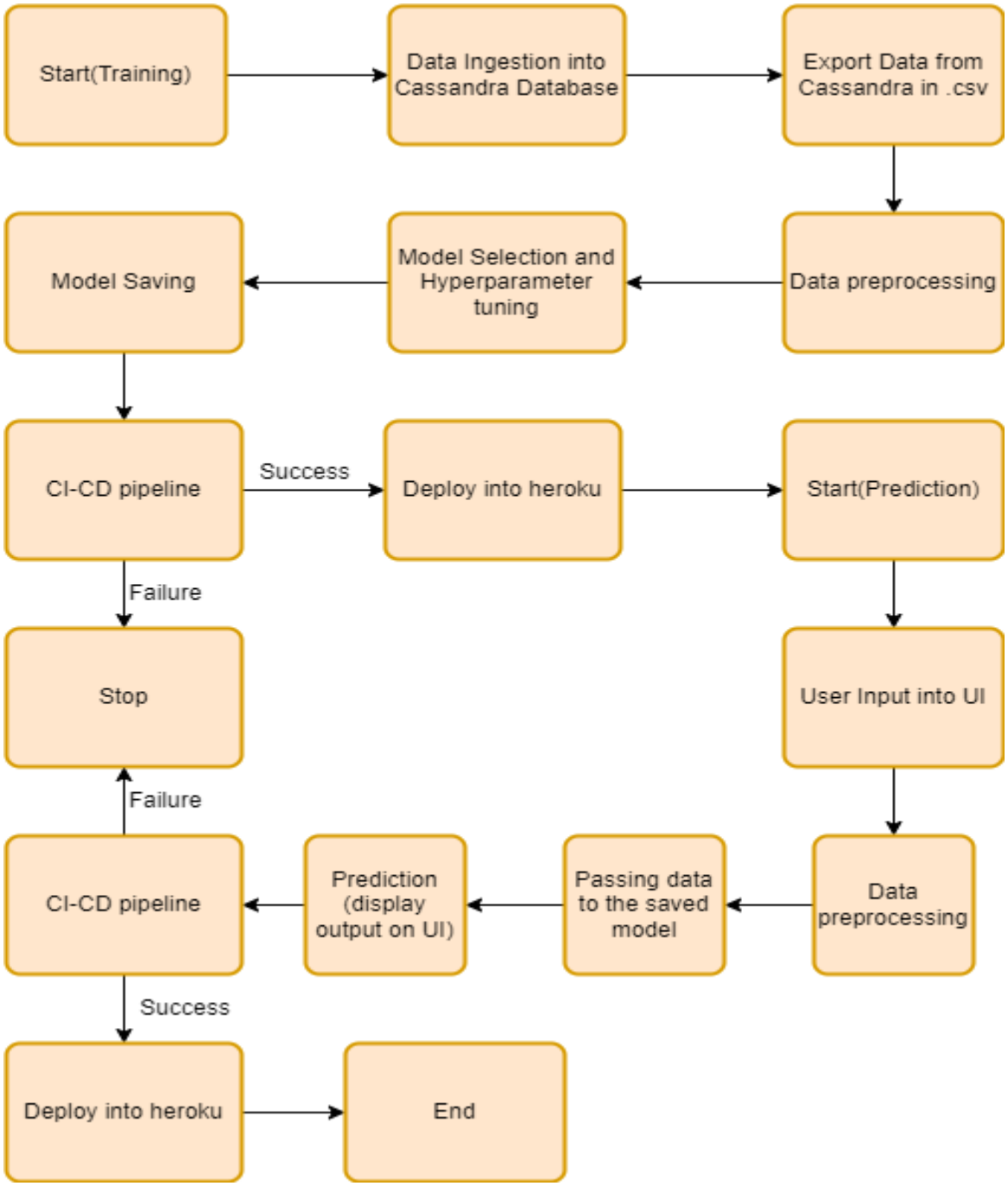
4.Technology stack

Front End	HTML/CSS/JS/Bootstrap
CI-CD pipeline	GitHub actions
Backend	Python Flask
Database	Cassandra
Deployment	heroku
Visualization	Sea born
Data version control	DVC
Source version control	GitHub

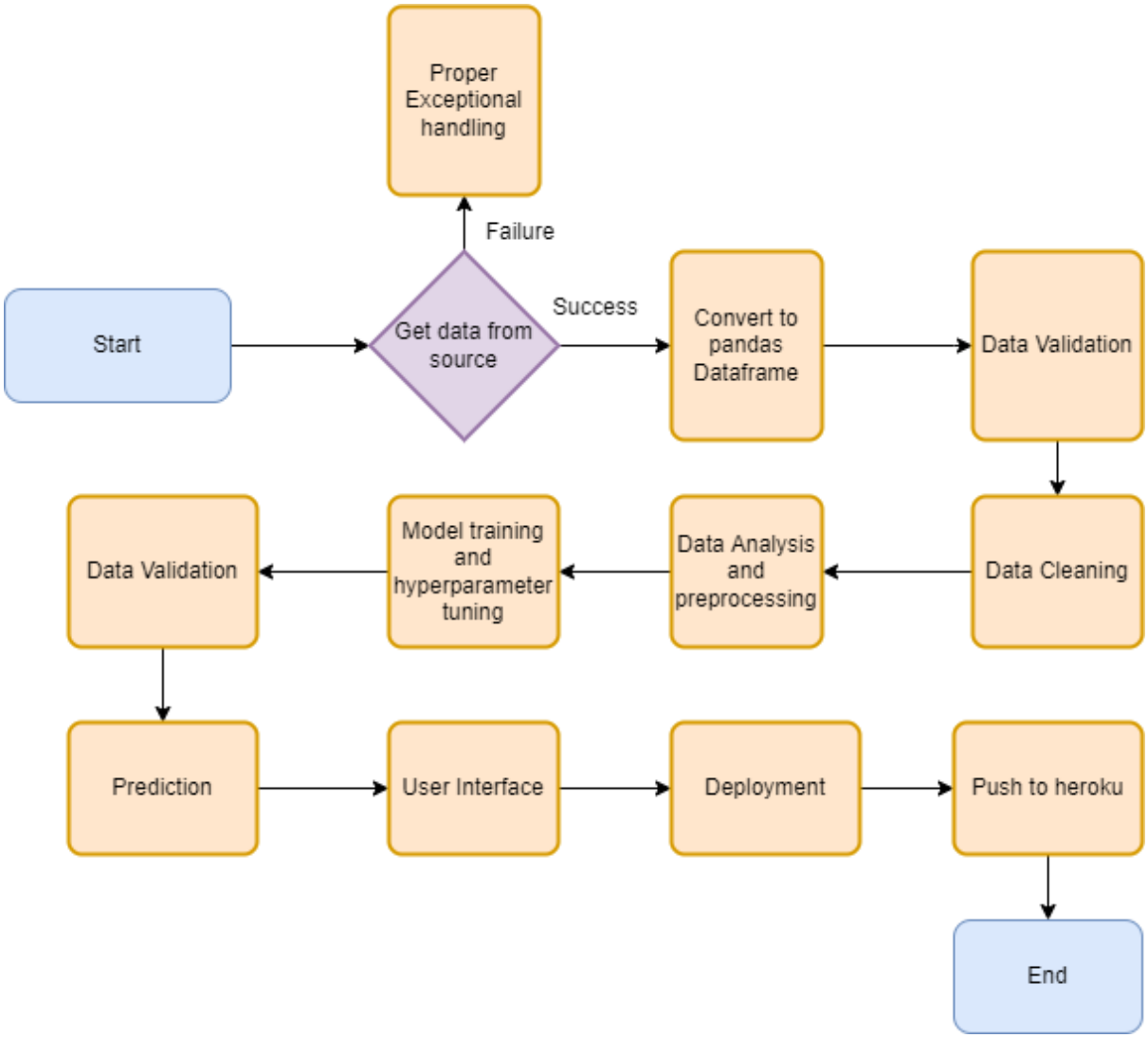
5. Proposed Solution

The solution proposed here is a Phishing domain detection based machine learning system to tackle phishing. In these approach we perform some classical machine learning task like Data Exploration, Data Cleaning, Feature Exploration and Feature Selection. Then building multiple models using different parameters and testing the same and selecting the best model giving good performance metrics. Then creating 2 flask APIs for training and prediction respectively and binding them with a frontend created using HTML, CSS, and Bootstrap. Finally hosting this solution on heroku by dockerizing complete module.

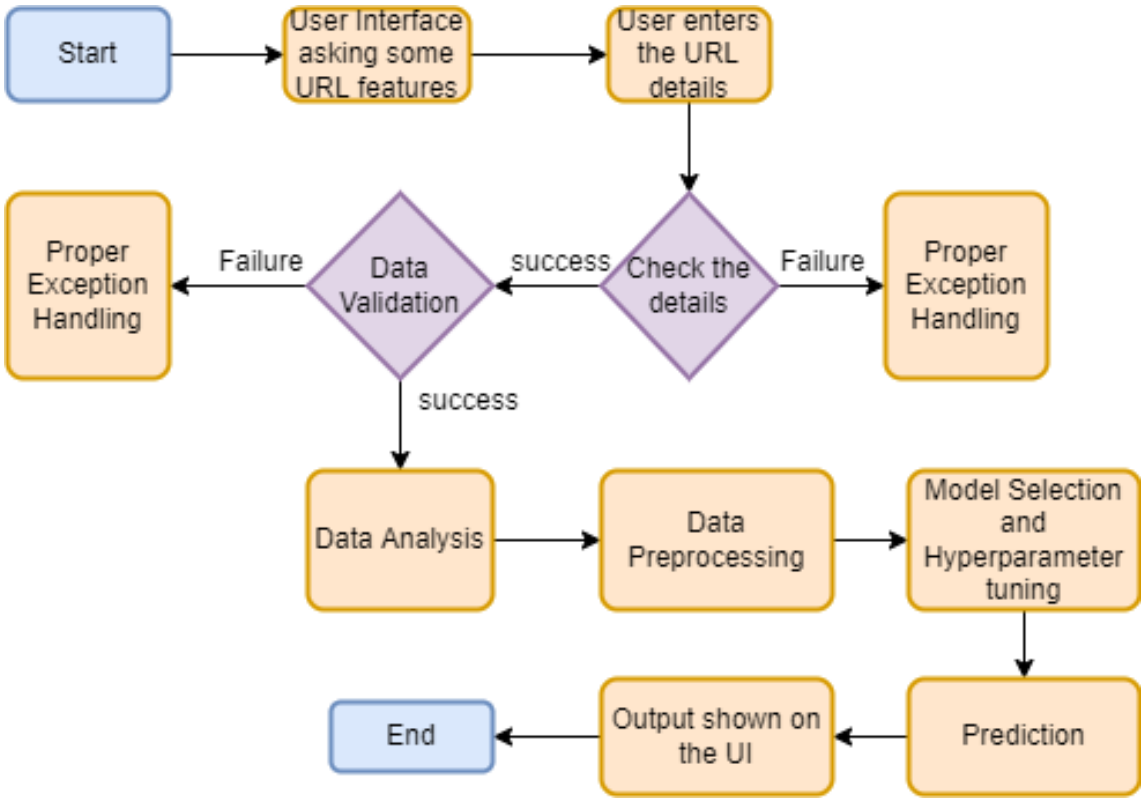
6. Proposed Architecture



7. Model training/validation workflow



8.User I/O workflow



9. Error Handling

Should errors be encountered, an explanation will be displayed as to what went wrong?

An error will be defined as anything that falls outside the normal and intended usage.

A python function is created to display the error message in detail.

The format of message will be current date -- current time -- error message.

10. Model performances

Use case (Binary Classification)	ML Model	Accuracy
1.	Logistic Regression	92.5%
2.	Random Forest	95%
3.	Support Vector Classifier	93.8%
4.	Naïve Bayes Classifier	92.1%
5.	Decision Tree Classifier	94.4%
6.	XGBoost Classifier	95.1%

11. Key performance indicators (KPI)

- Key indicators displaying a summary of the anomaly detection in the URLs.
- Total number of slash in URL
- Total number of characters in URL
- Total number of dots in URL domain section
- Total number of dots in URL directory section
- Total number of hyphen in URL directory section
- Total number of filename characters
- Total number of underlines in directory section of URL
- ASN IP number of URL
- Enter no of days of time domain activation
- Enter no of days of time domain expiration
- Enter time to live

12. Conclusion

The Phishing Domain Detection System will detect whether the domains is real for fake and avoid user to become a victim of phishing based on various data used to train our algorithm, so we can identify the fake domain(URL) so prevent the loss of crucial data by avoiding phishing.

13. References

1) Datasets for phishing websites detection - ScienceDirect
2) Phishing Websites Dataset - Mendeley Data
3) https://getbootstrap.com/docs/4.3/getting-started/introduction/
4) https://www.youtube.com/watch?v=1BSwYlJUxK0&list=PLZoTAELRMXVOk1pRcOCaG5xtXxgMalpIe
5) https://www.youtube.com/watch?v=ioN1jcWxbv8&list=PLZoTAELRMXVPQyArDHyQVjQxjj_YmEuO9
6) https://www.youtube.com/watch?v=uMlU2JaiOd8&list=PLZoTAELRMXVPgjwJ8VyRoqmfNs2CJwhVH
7) https://www.youtube.com/user/krishnaik06/playlists