Phishing Domain Detection

High Level Design (HLD)

Ramchandra Tukaram Padwal

Revision Number: 2.0

Last date of revision: 28/05/2023

Document Version Control

Date Issued	Version	Description	Author
02/05/2023	1	Initial HLD – V1.0	Ramchandra
28/05/2023	2	Final HLD –V2.0	Ramchandra

Contents

Abstract

Phishing stands for a fraudulent process, where an attacker tries to obtain sensitive information from the victim. Usually, these kinds of attacks are done via emails, text messages, or websites. Phishing websites, which are nowadays in a considerable rise, have the same look as legitimate sites. However, their backend is designed to collect sensitive information that is inputted by the victim. Discovering and detecting phishing websites has recently also gained the machine learning community's attention, which has built the models and performed classifications of phishing websites.

Introduction

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 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

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.

Definitions

Term	*Description*
CSV	Comma Separated Value
IDE	Integrated Development Environment
EDA	Exploratory Data Analysis
AWS	Amazon Web Services

General Description

Product Perspective

The phishing domain detection solution is a machine learning based domain detection model which will help us to detect the comparison between phishing sites and legitimate sites.

Problem statement

Phishing is a type of fraud in which an attacker impersonates a reputable company or person 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 is to predict whether the domains are real or malicious.

PROPOSED SOLUTION

The solution proposed here is a website interface where we will input the URL of the website into a field to check if the URL is fake and malicious. Checking and detection goes with the help of machine learning algorithms.

FURTHER IMPROVEMENTS

UGV can be added with more use cases like weather detection, live temperature to detect and record the temperature of that locality. UGV can also be synchronized with UAV (Unmanned Aerial Vehicle) for better and fast response or action, with help of UGV and UAV synchronization it can be implemented in the other domains like mining, agriculture.

The Phishing domain detection will mostly integrate with an android app where we will make a widget to detect phishing URLs when we click on incoming social app messages, for example whenever we get scam offer links in WhatsApp.

Technical Requirements

This document addresses the requirement for detecting the malicious phishing websites and alert the user before any type of loss will happen.

- For full functionality it must have a good resource server with minimum 2GB ram, 2 vcpu and above 10GB storage.
- Minimum server requirement with resources 1GB ram and 8GB storage and 1 vcpu for detection working

Data Requirements

Data requirements completely depend on our problem statement.

- We need URLs data which have a dataset with a mixture of real and malicious.
- URLs dataset also has a variety of URLs content with or without directory, file, params.

Tools used

Python programming language and frameworks such as NumPy, Pandas, Scikit-learn, TensorFlow, Keras and Roboflow are used to build the whole model.

- Visual Studio is used as an IDE.
- AWS elastic bean is used for deployment of the model.
- The Jupyter notebook is used for EDA.
- CSV is used to retrieve, insert some information.
- Front end development is done using HTML/CSS
- Python flask is used for backend development.
- GitHub is used as a version control system.

Hardware Requirements

- Server or for local is PC

Constraints

The phishing domain detection solution website must be user friendly, as automated as possible and users should not be required to know any of the workings.

Assumptions

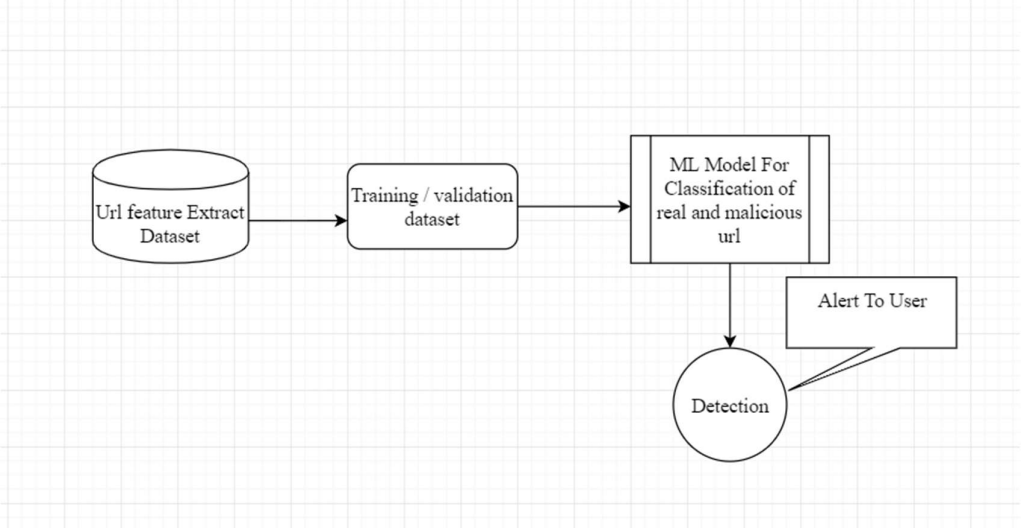
Assumption when we extract data from url if we did not get a specific feature of url then we will define it with -1 in data.

Design Details

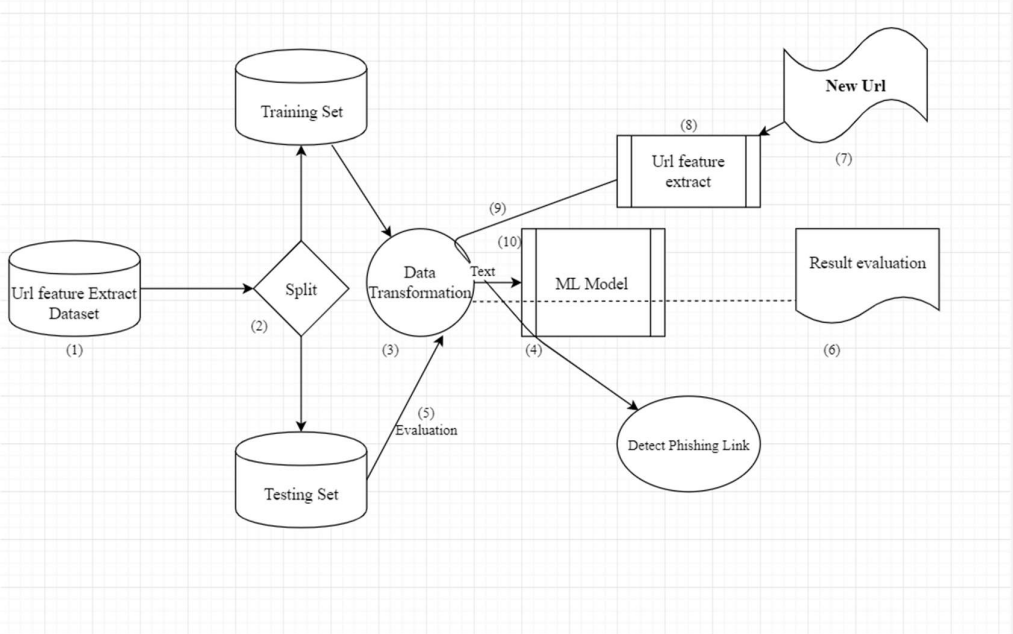
Process Flow

For identifying the different types of URL, we will use a machine learning base model. Below is the process flow diagram as shown below.

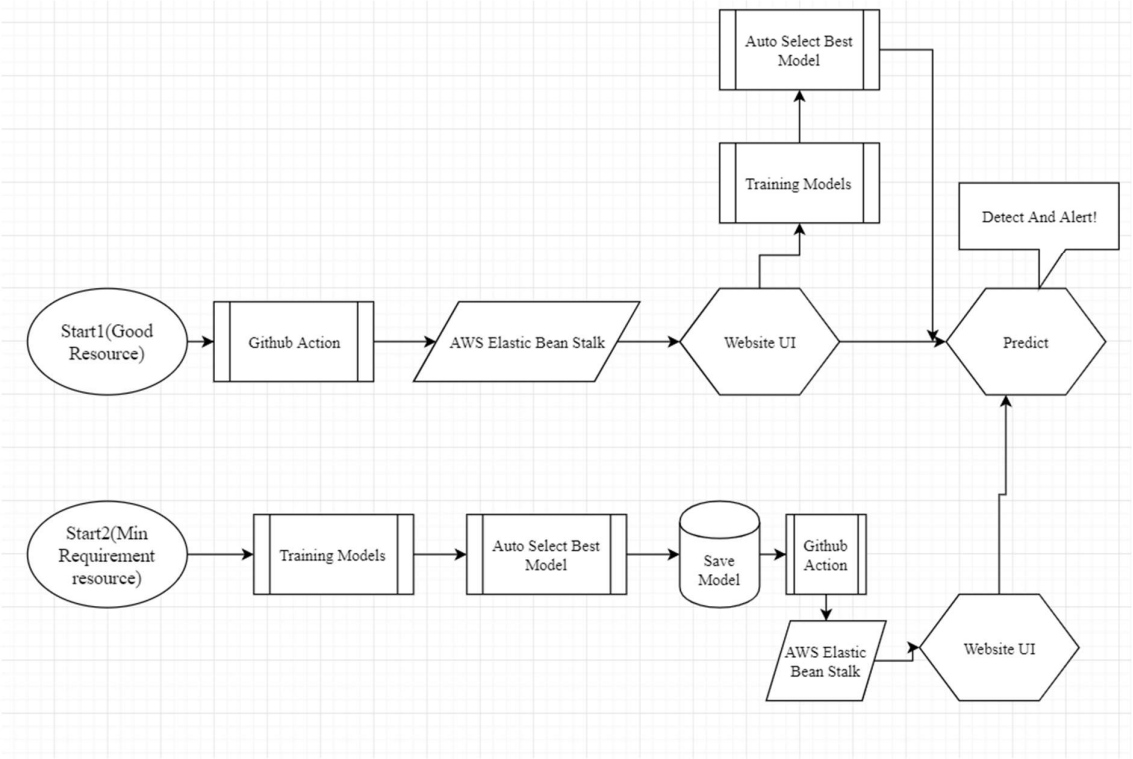
Proposed methodology



Model Training and Evaluation



Deployment Process



Event log

The system should log every event so that the user will know what process is running internally.

Initial Step-By-Step Description:

1. The System identifies at what step logging required
2. The System should be able to log each and every system flow.
3. System should not hang even after using so many loggings. Logging just because we can easily debug issues, so logging is mandatory to do.

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.

Performance

This phishing domain detection is detecting a malicious vs real website So that it will not mislead the user. Timely make updates on training dataset whenever new false detection occurs. Also, model retraining is very important to improve the performance.

Reusability

The code written and the components used should have the ability to be reused with no problems.

Application Compatibility

The different components for this project will be using Python as an interface between them. Each component will have its own task to perform, and it is the job of the Python to ensure proper transfer of information.

Resource Utilization

When any task is performed, it will likely use all the processing power and ram space available until that function is finished.

Deployment

Dashboards

Dashboards will be implemented to display and indicate certain KPIs and relevant indicators for the unveiled problems that if not addressed in time could cause catastrophes of unimaginable impact.

As and when, the system starts to capture the historical/periodic data for a user, the dashboards will be included to display charts over time with progress on various indicators or factors.

KPIs (Key Performance Indicators)

1. Key indicators displaying a summary of the dataset.

Conclusion

The phishing domain detection will detect a malicious website based on machine learning models . After this user will be safe from any loss or safe internet browsing.

References

6. https://www.sciencedirect.com/science/article/pii/S2352340920313202
7. Google.com for images of tools.
8. External test URL dataset for checking accuracy
   https://www.kaggle.com/datasets/shashwatwork/web-page-phishing-detection-dataset