Low Level Design (LLD)

Phishing Attack Domain Detection

Revision Number: 1
Last date of revision: 15/12/2021

Deepesh Mhatre

• Document Version Control

Date Issued	Version	Description	Author
15 Dec 2021	1.0	First Draft	Deepesh Mhatre

Contents

Abstract

We live in the digital era. This has steadily changed the way you buy things, pay your bills, rent an apartment, watch a movie, and everything else. All of this is made possible, because of the internet and electronic media.Due to this widespread use, there are innumerable incidents of a security breach, fraud,malicious attacks, etc reported. To keep the internet age well-ordered and safe for users, the need for Cybersecurity arises. It secures you from Cyber-criminals, fraudsters, hackers, and anybody who wants to harm you financially, mentally, or engage in data theft online.

This project focuses on solving a most common cyber fraud called phishing.Phishing is a type of social engineering attack often used to steal user data, including login credentials and credit card numbers. It occurs when an attacker, masquerading as a trusted entity, dupes a victim into opening an email, instant message, or text message. The recipient is then tricked into clicking a malicious link, which can lead to the installation of malware, the freezing of the system as part of a ransomware attack or the revealing of sensitive information.In this project we discuss about creating a machine learning system which can detect malicious url links and how we can prevent phishing attacks using it.

- Serviceability

1.Introduction

1.1 Why this Low-Level Design Document?

The purpose of this Low-Level Design (LLD) 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 LLD 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

a. 1.2 Scope

The LLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The LLD 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 unusual activity , and fire disasters.

2.Technical specifications

a. 2.1 Dataset

• Cases	Finalized	Source
Malicious URL	yes	Google/kaggle
Legitimate URL	Yes	Google/kaggle

a. 2.2 Dataset overview

The dataset consist of 2 classes,malicious and legitimate URLs. It contains 450k domain urls out of which 345k are legitimate and 104k are malicious. The Imbalanced dataset is oversampled using the SOMTE technique,which increases the total number of samples to around 600k. we extract some useful features from these urls and further improve our dataset to make it more suitable for training ML models.The below mentioned category of features are extracted from the URL data :

- Length based Features (5 features)
- Count based Features (11 features)
- Binary Features (2 features)

b. 2.3 Input schema

Feature name	Datatype	Size	Null/Required
URL	String		Required

3. Deployment

1. AWS
2.
3.

• 4.Technology stack

Front End	React,CSS,JS
Backend	Python FastAPI
Database	MongoDB/MySql
Deployment	Heroku
Visualization	Matplotlib,Seaborn ,Plotly
Dashboard	Tableau/Power BI
version control	GitHub

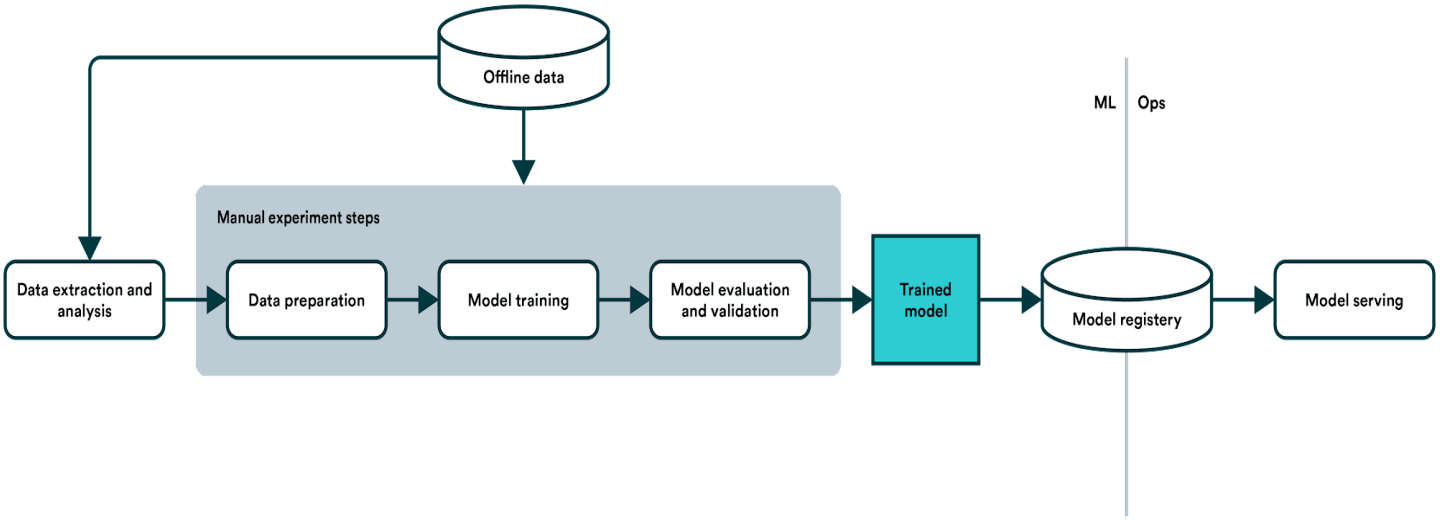
• 5. Proposed Solution

The above mentioned problem statement can be solved by Implementing a deep learning based regression model. On a high level the working of the model is simple.The model will take a URL as input and predict the probability of the URL being malicious.

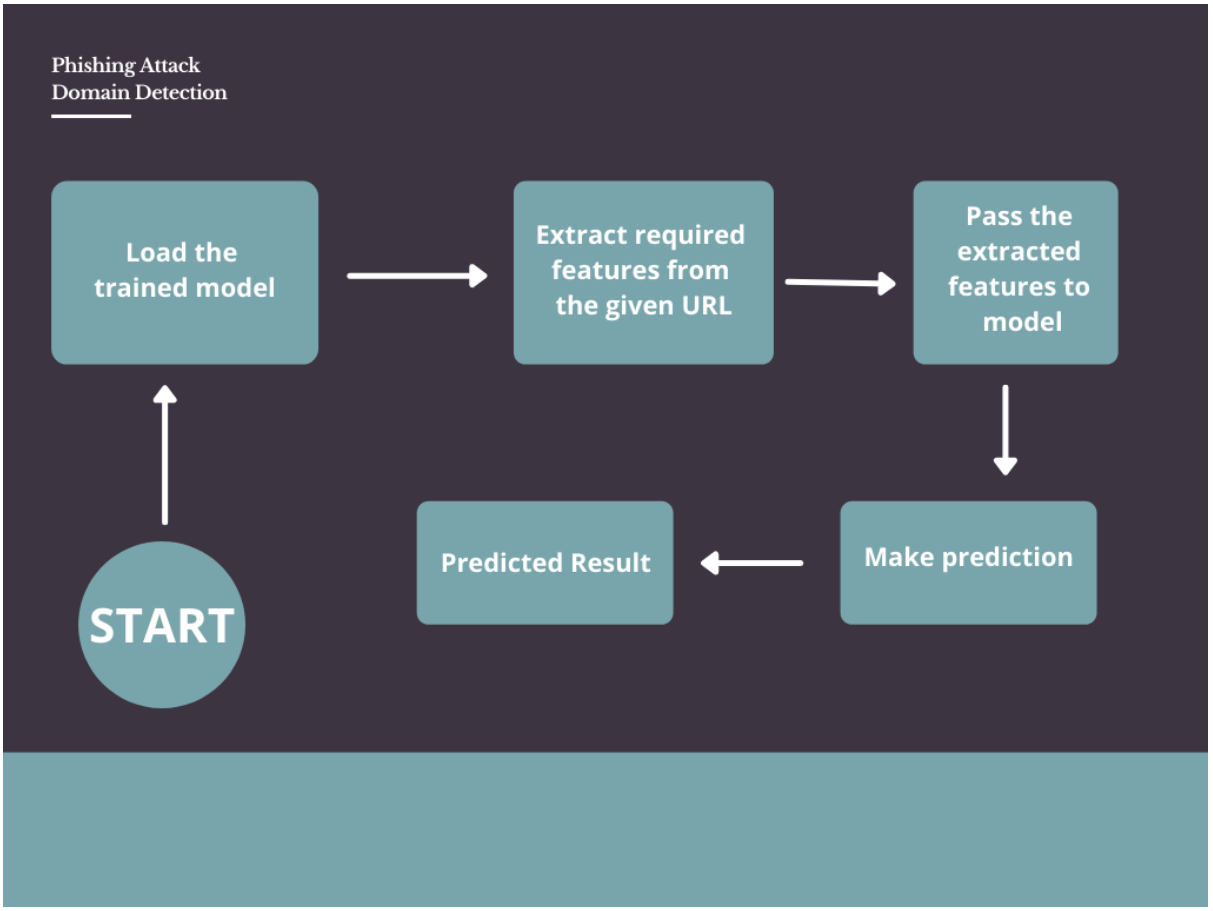
The model will look at various other features & characteristics of the domain to determine if the URL is malicious or legitimate.

1. Baseline Models : Decision Tree,Random Forest,Neural Network

6.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.

-

- ## 10.Test cases

Use case	Module	Accuracy
URL detection	Decision Tree	75%
URL detection	Random Forest	85%
URL detection	Neural Network	95%

12. Conclusion

The Designed deep-learning based Phishing domain detection system takes a string URL as input and returns a probability value (0-100) of URL being malicious. We declare a URL malicious if it crosses a probability value of 70%. It is deployed as both REST API and a web interface has also been created to interact with it.