

Wireframe

Phishing Domain Detection (Machine Learning)

By

Kavitha Narsapur

Overview

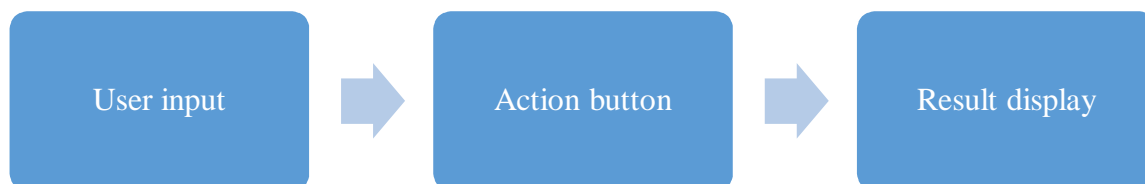
Phishing is a type of fraud in which an attacker impersonates a reputable company or person in order to get sensitive information such as login credentials or account information via email or other communication channels. Phishing is popular among attackers because it is easier to persuade someone to click a malicious link that appears to be authentic than it is to break through a computer's protection measures. Phishing attacks are done via emails, text messages, or websites. Phishing websites have the same look as legitimate sites. However, their backend is designed to collect sensitive information that is inputted by the victim.

This document describes the wireframe for the schematic design of web interface created for the machine learning model. We will also discuss how web interface will be connected to the machine learning model.

Web interface wireframe

A wireframe is a two dimensional illustration of a page's interface that specifically focuses on space allocation and prioritization of contents, functionalities available and intended behaviors.

The interface we have created consists of only a single page through which the user interacts with machine learning model. Below is the model for this project.



Component functions

Below are the roles that each component in wireframe performs.

1. User Input: Takes the user input from the user and prepares for transmission to ML API.
2. Action button: When clicked, it sends user input to Rest API as a POST request.
3. Result display: The component is responsible for displaying the result received from POST request to ML model.

User interaction

Previously we saw what function each component inside the wireframe performs. Now we will see how these components work together to facilitate communication between the user and the model.

The sequence in which communication happens:

1. User enters the URL inside the input box.
2. User clicks the action button and sends a POST request to API.
3. The displayer receives the output from the API and updates the result.