**Phishing Domain Detection Project**

**High-Level Design (HLD)**

Document Version:1.0

Control Version Date: 22-Feb-24

Author : Akshay kumar BM

Comments: Introduction & Architecture defined

**Contents**

Introduction

1. Why High Level Design
2. General Description
   1. Product Perspective
   2. Problem Statement
   3. Proposed Solution
   4. TechnicalRequirement5
   5. Data Requirements
   6. Tools Used
   7. Constraints
3. Design Details
   1. Process Flow
   2. Deployment Process
4. Performance
   1. Re- usability
   2. Application Compatibility

4.3. Resource Utili7ation

4.4. Deployment

4.5 User interface

1. Conclusion

**Abstract**

This High-Level Design (HLD) outlines the architectural blueprint and strategic approach for developing a machine learning-based credit card default prediction system. In the contemporary financial landscape, the ability to accurately forecast credit card defaults is indispensable for risk management and maintaining a healthy credit portfolio. The HLD encapsulates the various stages of the project lifecycle, commencing with the acquisition of diverse datasets encompassing historical credit card transactions, customer demographics, economic indicators, and potentially supplementary sources of data. Rigorous preprocessing techniques are applied to cleanse, engineer features, and standardize the datasets, priming them for subsequent modeling endeavors. Employing a supervised learning paradigm, the project explores an array of algorithms, including logistic regression, decision trees, random forests, gradient boosting, and neural networks, to construct predictive models. Algorithm selection is contingent upon factors such as performance metrics, interpretability, and computational efficiency. Validation and evaluation protocols are paramount to ascertain the efficacy and generalizability of the models. Techniques such as cross-validation, hyperparameter tuning, and comprehensive performance metrics evaluation are deployed to assess model performance rigorously. Following model development, extensive testing procedures are undertaken to validate the robustness and reliability of the predictive models across diverse real-world scenarios. This phase encompasses simulating various conditions and evaluating the model's performance under different circumstances. The deployment phase entails seamless integration of the trained model into the existing credit risk management infrastructure. The model may be deployed either as a standalone application or as an integral component within a broader decision-making framework. Continuous monitoring and iterative updates are pivotal to adapt to evolving market dynamics and sustain model efficacy over time

# High-Level Design Document for Phishing Website Detection System

## Introduction

The purpose of this document is to provide a high-level design (HLD) for a phishing website detection system. This system is designed to identify and flag potential phishing websites, thereby enhancing internet security and user safety.

## 1. Why High-Level Design

High-level design provides an overview of a system, product, or service. It outlines the system’s main components, their functionalities, and their interactions. In the context of our phishing website detection system, the HLD will provide a comprehensive view of how the system will function and interact with users and other systems.

## 2. General Description

### 2.1. Product Perspective

The phishing website detection system is an independent product designed to integrate with existing web services and browsers. It aims to enhance user safety during internet navigation by identifying and flagging potential phishing websites.

### 2.2. Problem Statement

Phishing attacks pose a significant threat to internet users. These attacks often involve deceptive websites that mimic legitimate ones to trick users into revealing sensitive information. There is a need for an effective solution to detect and flag these phishing websites.

### 2.3. Proposed Solution

The proposed solution is a machine learning-based system that can detect potential phishing websites based on various features of a website such as the URL structure, domain name, and other webpage elements.

### 2.4. Technical Requirements

The system requires a machine learning environment for model training and prediction. It also needs access to a database of website URLs for training the model. The system should be compatible with various operating systems and web browsers.

### 2.5. Data Requirements

The primary data required for this system is a large dataset of website URLs, both legitimate and phishing. Each URL should be labeled as either ‘phishing’ or ‘safe’, serving as the target variable for the machine learning model.

### 2.6. Tools Used

### 

The system is implemented using Python, with libraries such as pandas for data manipulation, sklearn for machine learning, and Flask for web application development.

### 2.7. Constraints

The system’s effectiveness is dependent on the quality and quantity of the training data. Additionally, while the system can accurately detect many phishing websites, no solution can guarantee 100% accuracy due to the continually evolving tactics employed by phishers.

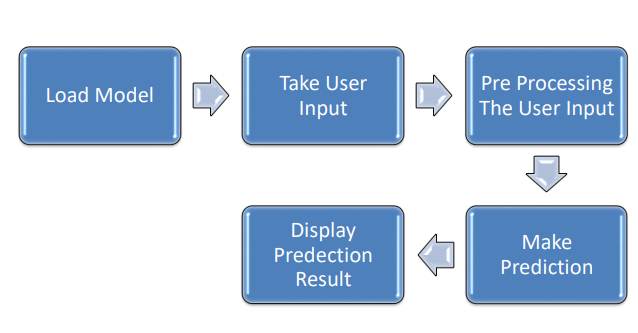
## 3. Design Details

### 3.1. Process Flow

### 

The system follows a three-step process: data ingestion, data transformation, and model training. During data ingestion, the system collects and stores website URLs. In the data transformation step, the system preprocesses the data, extracting features from the URLs. Finally, in the model training step, the system uses the preprocessed data to train a machine learning model.

### 3.2. Deployment Process

The trained model is then deployed as a web service using Flask. When a user navigates to a website, the system extracts features from the website’s URL and feeds them into the model, which then predicts whether the website is a potential phishing site. 

## 4. Performance

### 4.1. Reusability

The system’s code is modular and well-documented, promoting reusability. The data ingestion, data transformation, and model training components can be modified and reused for other machine learning tasks.

### 4.2. Application Compatibility

The system is designed to be compatible with various web browsers and can be integrated with other web services to enhance their security features.

### 4.3. Resource Utilization

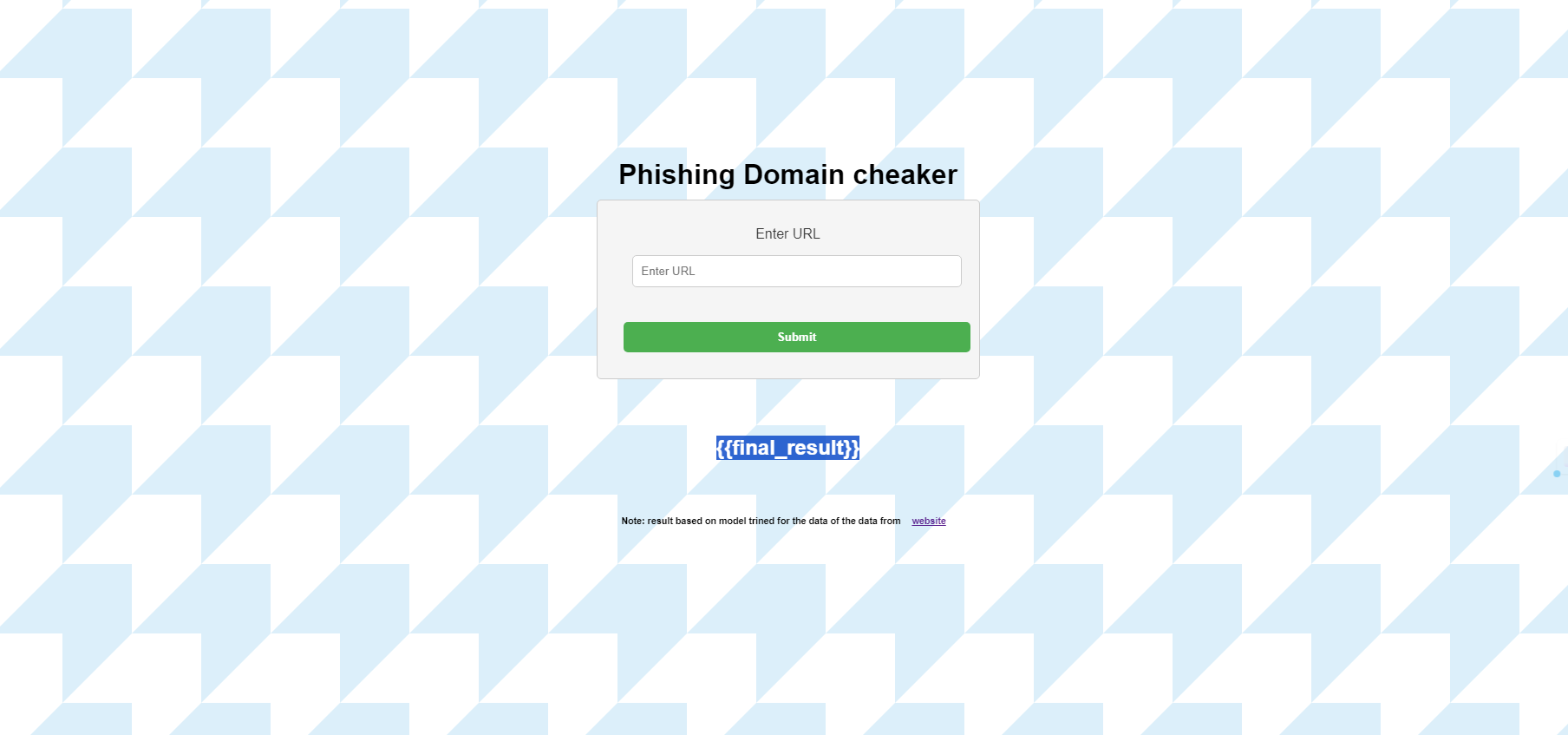
The system is optimized to minimize resource utilization. The model training process can be resource-intensive, but this is an offline process and does not affect the system’s runtime performance.

### 4.4. Deployment

The system is deployed as a web service, making it accessible to users on various platforms. The deployment process involves setting up a server, loading the trained model, and launching the web service.

### 4.5. User Interface

The system does not have a user interface in the traditional sense. Instead, it works in the background, analyzing website URLs and alerting users if a potential phishing website is detected.



## 5. Conclusion

The phishing website detection system is a robust and efficient solution to a prevalent internet security issue. By leveraging machine learning techniques, the system can accurately identify potential phishing websites, enhancing user safety during internet navigation. Despite certain constraints, the system promises to be a valuable tool in the ongoing fight against phishing attacks.