Phishing Domain Detection System Project Report

1. Introduction

The Phishing Domain Detection System aims to mitigate the increasing threat of cyber fraud. With the rise of online transactions and digital interactions, the risk of phishing attacks—where malicious actors impersonate legitimate entities to steal sensitive information—has become more prevalent. Our project leverages machine learning techniques to distinguish between authentic and fraudulent domains. This report provides an overview of the system's objectives, methodology, and outcomes.

---

2. System Requirements

2.1 Functional Requirements

Our system fulfills the following:

Domain Classification: Classify domains as safe or phishing using features extracted from URLs.

User Interface: Provide a simple web-based interface for users to test domain URLs.

2.2 Non-Functional Requirements

Code Quality: Maintain a clean, modular, and well-documented codebase for maintainability and ease of understanding.

User Accessibility: Ensure the interface is responsive and works reliably on standard systems.

---

3. Technology Stack

Programming Language: Python

Machine Learning: Scikit-learn

Web Framework: Flask

Interface: HTML + Flask for user input and results display

---

4. Data Flow

Data Acquisition: Dataset is downloaded from Mendeley (Phishing Websites Dataset).

Preprocessing: Cleaning and formatting of dataset to handle inconsistencies.

Feature Engineering: Extract features from URLs like IP presence, HTTPS, domain age, etc.

Model Training: Train models like Random Forest using Scikit-learn.

Model Evaluation: Use accuracy, precision, recall, and F1-score for assessment.

Model Deployment: Load the trained model locally for inference in Flask.

User Interaction: Users input a domain via the web form, and get a prediction instantly.

---

5. Project Overview

5.1 Problem Statement

The project combats phishing—a major cybercrime tactic—by detecting fake domains that imitate legitimate ones.

5.2 Dataset

We used the publicly available **Phishing Websites Dataset** from Mendeley, which includes labeled examples of both legitimate and phishing domains.

5.3 Approach

Our method includes:

Cleaning and preprocessing data

Feature extraction (e.g., URL length, domain age, SSL presence)

Model training and evaluation

Using a **Random Forest** classifier for final prediction due to its strong performance on structured data

5.4 Technologies Used

Python (Main language)

Scikit-learn (Machine learning)

Flask (Backend + routing)

HTML/CSS (Frontend)

---

6. Conclusion

The Phishing Domain Detection System provides a lightweight and efficient tool for identifying phishing domains using machine learning. By removing the need for databases, logging, and cloud infrastructure, it becomes highly portable and easy to deploy on any local machine. The intuitive interface makes it accessible to users with little technical expertise, contributing to safer web interactions.