**Architecture Diagram**

**Phishing Domain Detection**

( **Machine Learning Project** )

Anshika Solanki

# 

# Document Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| Date Issued | Version | Description | Author |
| 7th Sep 2024 | 1.1 | Initial ADD – V1.0 | Anshika Solanki |

**Contents**

Document Version Control       2

Abstract                   4

1. Introduction       5

1.1 Why this Low-Level Design Document? 5

1.2 Scope 6

1.3 Constraints 7

1.4 Risks 8

1.5 Out of Scope 8

 2. Technical specifications                                                                                 8

* 1. Dataset 9
     1. Dataset Overview 9
     2. Input Schema 9

 2.2 Phishing Domain Detection Algorithm 13

* 1. Logging 17
  2. Database               18

3. Technology stack     18

4. Proposed Solution 19

 4.1 System Architecture 21

 4.2 Data Flow Diagram (DFD)     22

5. Model training/validation workflow     23

6. User Input/Output Workflow 24

7. Exceptional Scenarios 25

**Abstract**

The **Phishing Domain Detection** project focuses on identifying and stopping phishing websites using machine learning. Phishing attacks are a big problem in cybersecurity, as they trick people into giving away sensitive information like passwords and credit card details. This project aims to build a system that can tell the difference between real websites and phishing sites by analysing various features of domain names.

The system works by extracting specific characteristics from domain names. These features are then used to train machine learning models that can accurately classify domains as either safe or dangerous. The goal is to create a detection system that is both accurate and efficient, with minimal false alarms.

This document outlines the detailed design of the project, including the technical specifications, how the system is structured, and the steps involved in training and testing the model. It serves as a guide for anyone involved in building and implementing the phishing detection system, providing clear instructions and information to ensure its successful development.

# Introduction

## Why this Architecture Design Document?

An **Architecture Design Document (ADD)** for a **Phishing Domain Detection** system is crucial for several reasons, especially in a project centred on machine learning (ML) for cybersecurity. Here's why creating this document is essential:

1. **Clarifying Structure**: It outlines the system's components (data collection, preprocessing, model training, etc.), ensuring a clear understanding of how everything fits together.
2. **Facilitating Collaboration**: Provides a reference for all team members, improving communication and alignment on the project's goals.
3. **Defining Data Flow**: Specifies how data moves through the system, ensuring efficient processing and real-time detection.
4. **Ensuring Scalability**: Anticipates performance needs and prepares the system to handle large datasets efficiently.
5. **Blueprint for Implementation**: Guides the development process by detailing tools, technologies, and methods for each part of the system.
6. **Supporting Security and Compliance**: Ensures that security measures and regulatory requirements are considered throughout the system.
7. **Future Maintenance**: Provides a roadmap for updates and scalability as the system evolves.

In short, it ensures that the system is well-structured, scalable, and secure, making the development process smoother and more efficient.Bottom of Form

## Scope

The **scope of architecture** in a **Phishing Domain Detection** system includes:

1. **System Overview**: Defines the purpose and objectives of detecting phishing domains using machine learning.
2. **Key Components**:
   * **Data Sources**: DNS logs, WHOIS records, web page content.
   * **Data Preprocessing**: Cleaning and feature extraction.
   * **Feature Engineering**: Extracting relevant phishing indicators (e.g., domain age).
   * **Model Training**: ML algorithms to classify domains.
   * **Detection**: Real-time identification and alerts for phishing threats.
3. **Data Flow**: Details how data moves from collection to model prediction.
4. **Technology Stack**: Lists the tools and technologies used for data processing and model deployment.

## Constraints

* **Data Availability** : Limited or incomplete domain data.
* **Data Quality** : Noisy or unreliable data affecting accuracy.
* **Real-Time Performance** : Need for fast processing and detection.
* **Scalability** : Ability to handle growing volumes of data.
* **Computational Resources** : High demands for processing and inference.
* **Model Accuracy** : Balancing precision and recall to avoid false positives and negatives.
* **Security** : Protecting against adversarial attacks and ensuring system integrity.

## Out of Scope

* **Advanced Threat Intelligence**: No integration with external threat intelligence platforms.
* **Full Cybersecurity Suite**: Excludes other cybersecurity features like firewalls or antivirus.
* **User Behavior Analysis**: Does not include monitoring user activities or interactions.
* **Content Analysis**: Focuses only on domain attributes, not on-site content like images or text.
* **Mobile Phishing Detection**: Does not cover phishing threats targeting mobile apps.
* **Multi-Language Support**: No support for phishing detection in multiple languages or regions.
* **Historical Analysis**: No analysis of past phishing trends, only current domain detection.
* **Incident Response**: Does not involve actions or responses to phishing incidents.
* **Domain Registration Monitoring**: Does not monitor new domain registrations, only existing domains.
* **User Training**: No educational programs for users on phishing awareness.

# Technical specifications

## 2.1 Dataset

|  |  |  |
| --- | --- | --- |
| **Name** | **Finalized** | **Source** |
| Phishing Dataset | yes | <https://data.mendeley.com/datasets/72ptz43s9v/1> |

## 2.1.1 Dataset overview

The presented dataset was collected and prepared for the purpose of building and evaluating various classification methods for the task of detecting phishing websites based on the uniform resource locator (URL) properties, URL resolving metrics, and external services.

The attributes of the prepared dataset can be divided into six groups:

* Table 1. Dataset attributes based on URL.

| **Nr.** | **Attribute** | **Format** | **Description** | **Values** |
| --- | --- | --- | --- | --- |
| 1 | qty\_dot\_url | Number of ”.” signs | Numeric |  |
| 2 | qty\_hyphen\_url | Number of ”-” signs | Numeric |  |
| 3 | qty\_underline\_url | Number of ”\_” signs | Numeric |  |
| 4 | qty\_slash\_url | Number of ”/” signs | Numeric |  |
| 5 | qty\_questionmark\_url | Number of ”?” signs | Numeric |  |
| 6 | qty\_equal\_url | Number of ”=” sings | Numeric |  |
| 7 | qty\_at\_url | Number of ”@” signs | Numeric |  |
| 8 | qty\_and\_url | Number of ”&” signs | Numeric |  |
| 9 | qty\_exclamation\_url | Number of ”!” signs | Numeric |  |
| 10 | qty\_space\_url | Number of ” ” signs | Numeric |  |
| 11 | qty\_tilde\_url | Number of ″′˜′signs | Numeric |  |
| 12 | qty\_comma\_url | Number of ”,” signs | Numeric |  |
| 13 | qty\_plus\_url | Number of ”+” signs | Numeric |  |
| 14 | qty\_asterisk\_url | Number of ”\*” signs | Numeric |  |
| 15 | qty\_hashtag\_url | Number of ”#” signs | Numeric |  |
| 16 | qty\_dollar\_url | Number of ”$” signs | Numeric |  |
| 17 | qty\_percent\_url | Number of ”%” signs | Numeric |  |
| 18 | qty\_tld\_url | Top level domain character length | Numeric |  |
| 19 | length\_url | Number of characters | Numeric |  |
| 20 | email\_in\_url | Is email present | Boolean | [0, 1] |

* Table 2. Dataset attributes based on domain URL.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | **Nr.** | **Attribute** | **Format** | **Description** | **Values** | | --- | --- | --- | --- | --- | | 1 | qty\_dot\_domain | Number of ”.” signs | Numeric |  | | 2 | qty\_hyphen\_domain | Number of ”-” signs | Numeric |  | | 3 | qty\_underline\_domain | Number of ”\_” signs | Numeric |  | | 4 | qty\_slash\_domain | Number of ”/” signs | Numeric |  | | 5 | qty\_questionmark\_domain | Number of ”?” signs | Numeric |  | | 6 | qty\_equal\_domain | Number of ”=” signs | Numeric |  | | 7 | qty\_at\_domain | Number of ”@” signs | Numeric |  | | 8 | qty\_and\_domain | Number of ”&” signs | Numeric |  | | 9 | qty\_exclamation\_domain | Number of ”!” signs | Numeric |  | | 10 | qty\_space\_domain | Number of ” ” signs | Numeric |  | | 11 | qty\_tilde\_domain | Number of ”signs | Numeric |  | | 12 | qty\_comma\_domain | Number of ”,” signs | Numeric |  | | 13 | qty\_plus\_domain | Number of ”+” signs | Numeric |  | | 14 | qty\_asterisk\_domain | Number of ”\*” signs | Numeric |  | | 15 | qty\_hashtag\_domain | Number of ”#” signs | Numeric |  | | 16 | qty\_dollar\_domain | Number of ”$” signs | Numeric |  | | 17 | qty\_percent\_domain | Number of ”%” signs | Numeric |  | | 18 | qty\_vowels\_domain | Number of vowels | Numeric |  | | 19 | domain\_length | Number of domain characters | Numeric |  | | 20 | domain\_in\_ip | URL domain in IP address format | Boolean | [0, 1] | | 21 | server\_client\_domain | ”server” or ”client” in domain | Boolean | [0, 1] | |

* Table 3. Dataset attributes based on URL directory.

| **Nr.** | **Attribute** | **Format** | **Description** | **Values** |
| --- | --- | --- | --- | --- |
| 1 | qty\_dot\_directory | Number of ”.” signs | Numeric |  |
| 2 | qty\_hyphen\_directory | Number of ”-” signs | Numeric |  |
| 3 | qty\_underline\_directory | Number of ”\_” signs | Numeric |  |
| 4 | qty\_slash\_directory | Number of ”/” signs | Numeric |  |
| 5 | qty\_questionmark\_directory | Number of ”?” signs | Numeric |  |
| 6 | qty\_equal\_directory | Number of ”=” signs | Numeric |  |
| 7 | qty\_at\_directory | Number of ”@” signs | Numeric |  |
| 8 | qty\_and\_directory | Number of ”&” signs | Numeric |  |
| 9 | qty\_exclamation\_directory | Number of ”!” signs | Numeric |  |
| 10 | qty\_space\_directory | Number of ” ” signs | Numeric |  |
| 11 | qty\_tilde\_directory | Number of ”signs | Numeric |  |
| 12 | qty\_comma\_directory | Number of ”,” signs | Numeric |  |
| 13 | qty\_plus\_directory | Number of ”+” signs | Numeric |  |
| 14 | qty\_asterisk\_directory | Number of ”\*” signs | Numeric |  |
| 15 | qty\_hashtag\_directory | Number of ”#” signs | Numeric |  |
| 16 | qty\_dollar\_directory | Number of ”$” signs | Numeric |  |
| 17 | qty\_percent\_directory | Number of ”%” signs | Numeric |  |
| 18 | directory\_length | Number of directory characters | Numeric |  |

* Table 4. Dataset attributes based on URL file name.

| **Nr.** | **Attribute** | **Format** | **Description** | **Values** |
| --- | --- | --- | --- | --- |
| 1 | qty\_dot\_file | Number of ”.” signs | Numeric |  |
| 2 | qty\_hyphen\_file | Number of ”-” signs | Numeric |  |
| 3 | qty\_underline\_file | Number of ”\_” signs | Numeric |  |
| 4 | qty\_slash\_file | Number of ”/” signs | Numeric |  |
| 5 | qty\_questionmark\_file | Number of ”?” signs | Numeric |  |
| 6 | qty\_equal\_file | Number of ”=” signs | Numeric |  |
| 7 | qty\_at\_file | Number of ”@” signs | Numeric |  |
| 8 | qty\_and\_file | Number of ”&” signs | Numeric |  |
| 9 | qty\_exclamation\_file | Number of ”!” signs | Numeric |  |
| 10 | qty\_space\_file | Number of ” ” signs | Numeric |  |
| 11 | qty\_tilde\_file | Number of ”signs | Numeric |  |
| 12 | qty\_comma\_file | Number of ”,” signs | Numeric |  |
| 13 | qty\_plus\_file | Number of ”+” signs | Numeric |  |
| 14 | qty\_asterisk\_file | Number of ”\*” signs | Numeric |  |
| 15 | qty\_hashtag\_file | Number of ”#” signs | Numeric |  |
| 16 | qty\_dollar\_file | Number of ”$” signs | Numeric |  |
| 17 | qty\_percent\_file | Number of ”%” signs | Numeric |  |
| 18 | file\_length | Number of file name characters | Numeric |  |

* Table 5. Dataset attributes based on URL parameters.

|  |
| --- |
|  |
| **Nr.** | | **Attribute** | **Format** | **Description** | **Values** |
| 1 | | qty\_dot\_params | Number of ”.” signs | Numeric |  |
| 2 | | qty\_hyphen\_params | Number of ”-” signs | Numeric |  |
| 3 | | qty\_underline\_params | Number of ”\_” signs | Numeric |  |
| 4 | | qty\_slash\_params | Number of ”/” signs | Numeric |  |
| 5 | | qty\_questionmark\_params | Number of ”?” signs | Numeric |  |
| 6 | | qty\_equal\_params | Number of ”=” signs | Numeric |  |
| 7 | | qty\_at\_params | Number of ”@” signs | Numeric |  |
| 8 | | qty\_and\_params | Number of ”&” signs | Numeric |  |
| 9 | | qty\_exclamation\_params | Number of ”!” signs | Numeric |  |
| 10 | | qty\_space\_params | Number of ” ” signs | Numeric |  |
| 11 | | qty\_tilde\_params | Number of ”signs | Numeric |  |
| 12 | | qty\_comma\_params | Number of ”,” signs | Numeric |  |
| 13 | | qty\_plus\_params | Number of ”+” signs | Numeric |  |
| 14 | | qty\_asterisk\_params | Number of ”\*” signs | Numeric |  |
| 15 | | qty\_hashtag\_params | Number of ”#” signs | Numeric |  |
| 16 | | qty\_dollar\_params | Number of ”$” signs | Numeric |  |
| 17 | | qty\_percent\_params | Number of ”%” signs | Numeric |  |
| 18 | | params\_length | Number of parameters characters | Numeric |  |
| 19 | | tld\_present\_params | TLD1present in parameters | Boolean | [0, 1] |
| 20 | | qty\_params | Number of parameters | Numeric |  |

* Table 6. Dataset attributes based on resolving URL and external services.

| **Nr.** | **Attribute** | **Format** | **Description** | **Values** |
| --- | --- | --- | --- | --- |
| 1 | time\_response | Domain lookup time response | Numeric |  |
| 2 | domain\_spf | Domain has SPF 2 | Boolean | [0, 1] |
| 3 | asn\_ip | ASN 3 | Numeric |  |
| 4 | time\_domain\_activation | Domain activation time (in days) | Numeric |  |
| 5 | time\_domain\_expiration | Domain expiration time (in days) | Numeric |  |
| 6 | qty\_ip\_resolved | Number of resolved IPs | Numeric |  |
| 8 | qty\_nameservers | Number of resolved NS4 | Numeric |  |
| 9 | qty\_mx\_servers | Number of MX 5servers | Numeric |  |
| 10 | ttl\_hostname | Time-To-Live (TTL) | Numeric |  |
| 11 | tls\_ssl\_certificate | Has valid TLS 6/SSL 7certificate | Boolean | [0, 1] |
| 12 | qty\_redirects | Number of redirects | Numeric |  |
| 13 | url\_google\_index | Is URL indexed on Google | Boolean | [0, 1] |
| 14 | domain\_google\_index | Is domain indexed on Google | Boolean | [0, 1] |
| 15 | url\_shortened | Is URL shortened | Boolean |  |
| 16 | phishing | Is phishing website | Boolean | [0, 1] |

## 2.1.2 Input schema

| **Feature Name** | **Data Type** | **Size** | **Null/Required** |
| --- | --- | --- | --- |
| qty\_dot\_url | Integer | 4 bytes | Required |
| qty\_hyphen\_url | Integer | 4 bytes | Required |
| qty\_underline\_url | Integer | 4 bytes | Required |
| qty\_slash\_url | Integer | 4 bytes | Required |
| qty\_questionmark\_url | Integer | 4 bytes | Required |
| qty\_equal\_url | Integer | 4 bytes | Required |
| qty\_at\_url | Integer | 4 bytes | Required |
| qty\_and\_url | Integer | 4 bytes | Required |
| qty\_exclamation\_url | Integer | 4 bytes | Required |
| qty\_space\_url | Integer | 4 bytes | Required |
| qty\_tilde\_url | Integer | 4 bytes | Required |
| qty\_comma\_url | Integer | 4 bytes | Required |
| qty\_plus\_url | Integer | 4 bytes | Required |
| qty\_asterisk\_url | Integer | 4 bytes | Required |
| qty\_hashtag\_url | Integer | 4 bytes | Required |
| qty\_dollar\_url | Integer | 4 bytes | Required |
| qty\_percent\_url | Integer | 4 bytes | Required |
| qty\_tld\_url | Integer | 4 bytes | Required |
| length\_url | Integer | 4 bytes | Required |
| qty\_dot\_domain | Integer | 4 bytes | Required |
| qty\_hyphen\_domain | Integer | 4 bytes | Required |
| qty\_underline\_domain | Integer | 4 bytes | Required |
| qty\_slash\_domain | Integer | 4 bytes | Required |
| qty\_questionmark\_domain | Integer | 4 bytes | Required |
| qty\_equal\_domain | Integer | 4 bytes | Required |
| qty\_at\_domain | Integer | 4 bytes | Required |
| qty\_and\_domain | Integer | 4 bytes | Required |
| qty\_exclamation\_domain | Integer | 4 bytes | Required |
| qty\_space\_domain | Integer | 4 bytes | Required |
| qty\_tilde\_domain | Integer | 4 bytes | Required |
| qty\_comma\_domain | Integer | 4 bytes | Required |
| qty\_plus\_domain | Integer | 4 bytes | Required |
| qty\_asterisk\_domain | Integer | 4 bytes | Required |
| qty\_hashtag\_domain | Integer | 4 bytes | Required |
| qty\_dollar\_domain | Integer | 4 bytes | Required |
| qty\_percent\_domain | Integer | 4 bytes | Required |
| qty\_vowels\_domain | Integer | 4 bytes | Required |
| domain\_length | Integer | 4 bytes | Required |
| domain\_in\_ip | Boolean | 1 byte | Required |
| server\_client\_domain | Integer | 4 bytes | Required |
| qty\_dot\_directory | Integer | 4 bytes | Required |
| qty\_hyphen\_directory | Integer | 4 bytes | Required |
| qty\_underline\_directory | Integer | 4 bytes | Required |
| qty\_slash\_directory | Integer | 4 bytes | Required |
| qty\_questionmark\_directory | Integer | 4 bytes | Required |
| qty\_equal\_directory | Integer | 4 bytes | Required |
| qty\_at\_directory | Integer | 4 bytes | Required |
| qty\_and\_directory | Integer | 4 bytes | Required |
| qty\_exclamation\_directory | Integer | 4 bytes | Required |
| qty\_space\_directory | Integer | 4 bytes | Required |
| qty\_tilde\_directory | Integer | 4 bytes | Required |
| qty\_comma\_directory | Integer | 4 bytes | Required |
| qty\_plus\_directory | Integer | 4 bytes | Required |
| qty\_asterisk\_directory | Integer | 4 bytes | Required |
| qty\_hashtag\_directory | Integer | 4 bytes | Required |
| qty\_dollar\_directory | Integer | 4 bytes | Required |
| qty\_percent\_directory | Integer | 4 bytes | Required |
| directory\_length | Integer | 4 bytes | Required |
| qty\_dot\_file | Integer | 4 bytes | Required |
| qty\_hyphen\_file | Integer | 4 bytes | Required |
| qty\_underline\_file | Integer | 4 bytes | Required |
| qty\_slash\_file | Integer | 4 bytes | Required |
| qty\_questionmark\_file | Integer | 4 bytes | Required |
| qty\_equal\_file | Integer | 4 bytes | Required |
| qty\_at\_file | Integer | 4 bytes | Required |
| qty\_and\_file | Integer | 4 bytes | Required |
| qty\_exclamation\_file | Integer | 4 bytes | Required |
| qty\_space\_file | Integer | 4 bytes | Required |
| qty\_tilde\_file | Integer | 4 bytes | Required |
| qty\_comma\_file | Integer | 4 bytes | Required |
| qty\_plus\_file | Integer | 4 bytes | Required |
| qty\_asterisk\_file | Integer | 4 bytes | Required |
| qty\_hashtag\_file | Integer | 4 bytes | Required |
| qty\_dollar\_file | Integer | 4 bytes | Required |
| qty\_percent\_file | Integer | 4 bytes | Required |
| file\_length | Integer | 4 bytes | Required |
| qty\_dot\_params | Integer | 4 bytes | Required |
| qty\_hyphen\_params | Integer | 4 bytes | Required |
| qty\_underline\_params | Integer | 4 bytes | Required |
| qty\_slash\_params | Integer | 4 bytes | Required |
| qty\_questionmark\_params | Integer | 4 bytes | Required |
| qty\_equal\_params | Integer | 4 bytes | Required |
| qty\_at\_params | Integer | 4 bytes | Required |
| qty\_and\_params | Integer | 4 bytes | Required |
| qty\_exclamation\_params | Integer | 4 bytes | Required |
| qty\_space\_params | Integer | 4 bytes | Required |
| qty\_tilde\_params | Integer | 4 bytes | Required |
| qty\_comma\_params | Integer | 4 bytes | Required |
| qty\_plus\_params | Integer | 4 bytes | Required |
| qty\_asterisk\_params | Integer | 4 bytes | Required |
| qty\_hashtag\_params | Integer | 4 bytes | Required |
| qty\_dollar\_params | Integer | 4 bytes | Required |
| qty\_percent\_params | Integer | 4 bytes | Required |
| params\_length | Integer | 4 bytes | Required |
| tld\_present\_params | Boolean | 1 byte | Required |
| qty\_params | Integer | 4 bytes | Required |
| email\_in\_url | Boolean | 1 byte | Required |
| time\_response | Float | 8 bytes | Required |
| domain\_spf | Boolean | 1 byte | Required |
| asn\_ip | Integer | 4 bytes | Required |
| time\_domain\_activation | Float | 8 bytes | Required |
| time\_domain\_expiration | Float | 8 bytes | Required |
| qty\_ip\_resolved | Integer | 4 bytes | Required |
| qty\_nameservers | Integer | 4 bytes | Required |
| qty\_mx\_servers | Integer | 4 bytes | Required |
| ttl\_hostname | Integer | 4 bytes | Required |
| ttl\_ip | Integer | 4 bytes | Required |
| qty\_conn\_domain | Integer | 4 bytes | Required |
| qty\_conn\_hostname | Integer | 4 bytes | Required |
| qty\_conn\_ip | Integer | 4 bytes | Required |
| qty\_ssl\_cert | Integer | 4 bytes | Required |
| domain\_ssl\_exists | Boolean | 1 byte | Required |
| qty\_ssl\_matches\_domain | Integer | 4 bytes | Required |
| qty\_whois\_registered\_domain | Integer | 4 bytes | Required |
| qty\_whois\_registered\_ip | Integer | 4 bytes | Required |
| qty\_ttl\_valid | Integer | 4 bytes | Required |
| qty\_time\_domain\_expiration | Integer | 4 bytes | Required |
| qty\_total\_subdomains | Integer | 4 bytes | Required |
| qty\_files\_on\_server | Integer | 4 bytes | Required |
| qty\_cdn\_used | Integer | 4 bytes | Required |
| domain\_trusted\_certificate | Boolean | 1 byte | Required |
| ssl\_cert\_valid | Boolean | 1 byte | Required |
| domain\_redirects | Integer | 4 bytes | Required |
| time\_redirect\_activation | Float | 8 bytes | Required |
| qty\_domains\_redirect | Integer | 4 bytes | Required |
| is\_phishing | Boolean | 1 byte | Required |

## 2.2 Phishing Domain Detection Algorithm

#### 1. **Data Collection**

* **Input**: A dataset containing domain URLs with features related to their structure, and a label indicating whether they are phishing or legitimate.
* **Output**: Raw dataset with features and labels.

#### 2. **Data Preprocessing**

* **Step 1**: **Feature Extraction**
  + Extract features from raw URLs, domains, directories, files, and parameters.
  + Examples: qty\_dot\_url, length\_url, qty\_hyphen\_domain, etc.
* **Step 2**: **Handling Missing Values**
  + Impute or remove missing values as necessary.
* **Step 3**: **Normalization/Scaling**
  + Normalize or scale numerical features if needed.
* **Step 4**: **Data Splitting**
  + Split the dataset into training and test sets (e.g., 80% training, 20% testing).

#### 3. **Model Selection**

* **Step 1**: **Choose Classification Algorithm**
  + Example algorithms: Logistic Regression, Decision Trees, Random Forests, Gradient Boosting Machines, Support Vector Machines (SVM), Neural Networks.
* **Step 2**: **Train Model**
  + Fit the selected model to the training dataset.
* **Step 3**: **Hyperparameter Tuning**
  + Optimize model parameters using techniques such as Grid Search or Random Search.

#### **4. Model Evaluation**

* **Step 1**: **Evaluate Model Performance**
  + Use metrics such as accuracy, precision, recall, F1-score, and ROC-AUC.
* **Step 2**: **Cross-Validation**
  + Perform cross-validation to ensure the model generalizes well.

#### **5. Model Deployment**

* **Step 1**: **Integrate Model**
  + Deploy the model into the application or system where it will classify new domains.
* **Step 2**: **Real-Time or Batch Processing**
  + Ensure the model can handle real-time or batch classification as needed.

#### **6**. **Model Monitoring and Maintenance**

* **Step 1**: **Monitor Model Performance**
  + Track the performance of the model over time.
* **Step 2**: **Update Model**
  + Retrain or fine-tune the model with new data if performance degrades.
* **Step 3**: **Feature Reassessment**
  + Reevaluate features and consider incorporating new ones if necessary.

## 2.3 Logging

* Record every important action related to phishing detection, like data processing, model training, and predictions.
* The system should determine which steps need logging to ensure all critical processes are tracked.
* Capture the complete flow of data and actions through the system, from start to finish.
* Developers can pick between database logging or file logging, depending on what suits the system best.
* Logging should be done in a way that doesn’t slow down or hang the system.
* Logging is required to help with debugging and fixing issues, so it must be implemented throughout the system.

## 2.4 Database

* All incoming requests and data related to phishing detection should be stored in the database for future reference and analysis.
* Organize the stored data in a way that makes it easy to use for retraining the detection model whenever needed.

# Technology stack

|  |  |
| --- | --- |
| **Programming Language** | Python |
| **Data Preprocessing** | Numpy, Pandas |
| **Machine Learning** | Scikit Learn, RandomForest, Gradient Boost |
| **Database** | Sql |
| **Deployment** | Github |
| **Integrated Development Env.** | Jupyter Notebook |

# Proposed Solution

### **Objective**

To develop a machine learning model that can detect phishing domains based on various URL and domain features, ensuring accurate classification to protect users from phishing attacks.

### **Data Collection and Preparation**

**Dataset**: Utilize the provided dataset containing features related to URLs, domains, directories, files, and parameters. Key features include:

* + Frequency counts of special characters in the URL.
  + Domain-specific attributes.
  + File and directory information.
  + Additional features like SSL certificate status, DNS records, and response time.

**Data Cleaning**:

* + Handle missing values.
  + Normalize and standardize features as necessary.

**Feature Engineering**:

* + Extract relevant features from the raw dataset.
  + Create additional features if needed for improving model performance.

### **Model Development**

**Model Selection**:

* + Choose appropriate machine learning algorithms such as Random Forest, Gradient Boosting, or XGBoost based on their performance and suitability for classification tasks.

**Training and Validation**:

* + Split the dataset into training and validation sets.
  + Train multiple models and evaluate their performance using metrics like accuracy, precision, recall, F1-score, and ROC-AUC.

**Hyperparameter Tuning**:

* + Use techniques such as Grid Search or Random Search to find the optimal hyperparameters for the chosen models.

**Model Evaluation**:

* + Assess the model’s performance on a separate test set.
  + Ensure the model generalizes well and does not overfit.

### **Model Deployment**

**Database Management**:

* + Store user-submitted URLs and associated features in a MySQL database for future analysis and model retraining.

**Monitoring and Maintenance**:

* + Continuously monitor the model’s performance and update it with new data to maintain accuracy.

### **Model Retraining**

**Data Collection**:

* + Periodically collect new data from user interactions and stored in the database.

**Retraining**:

* + Use the updated dataset to retrain the model and improve its performance.

### **Security and Privacy**

**Data Protection**:

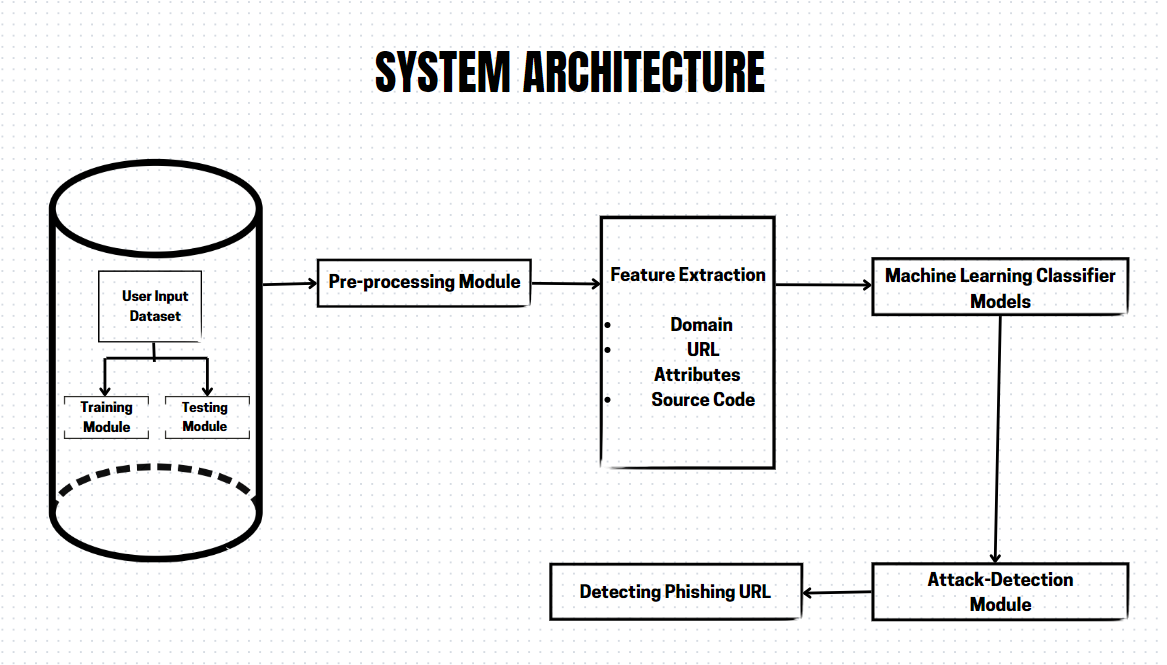
* + Ensure that all user data is stored securely and in compliance with relevant data protection regulations.

**Model Security**:

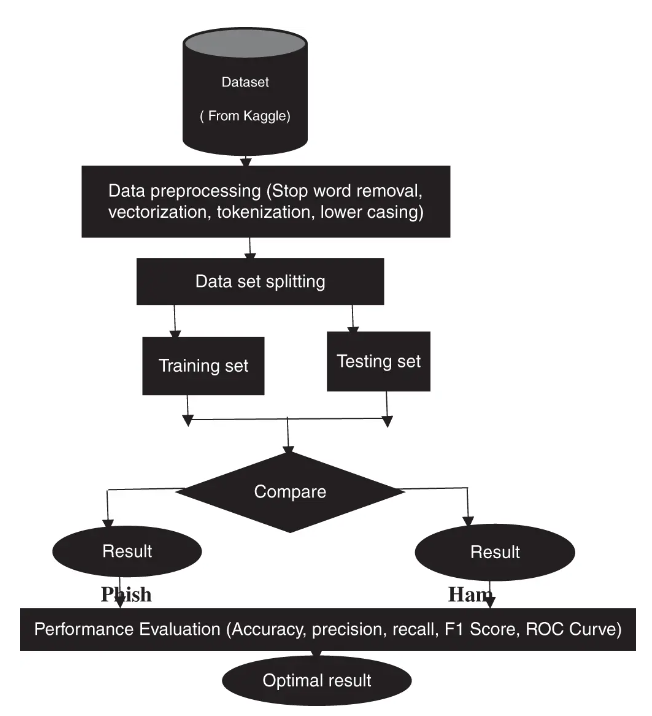
* + Implement measures to prevent unauthorized access to the model and its prediction

This solution outlines a comprehensive approach to developing and deploying a phishing domain detection model, from data preparation to model retraining and security considerations.

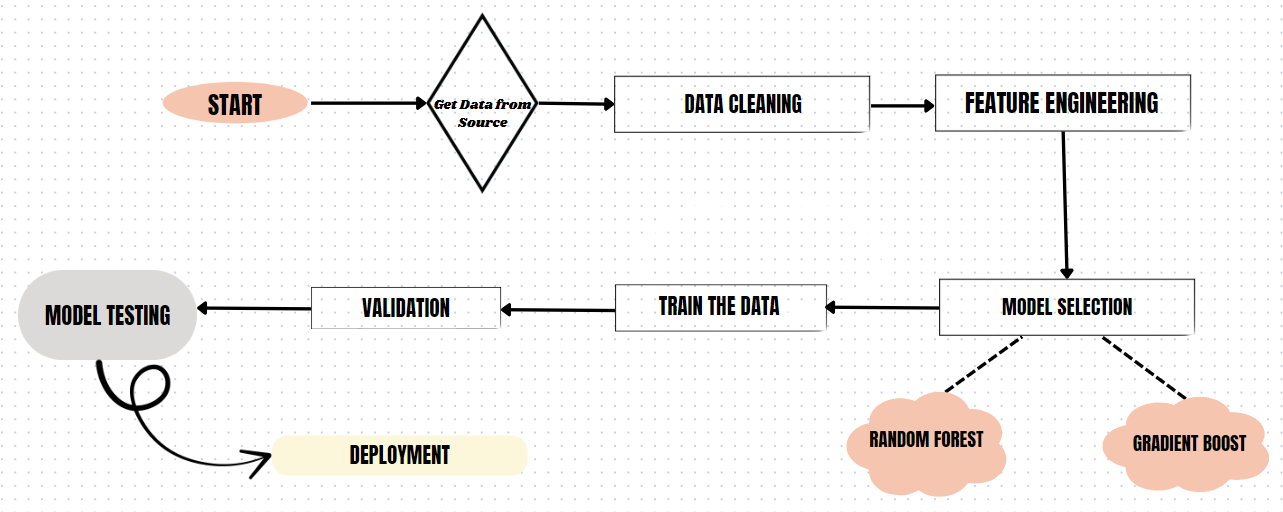
## System Architecture



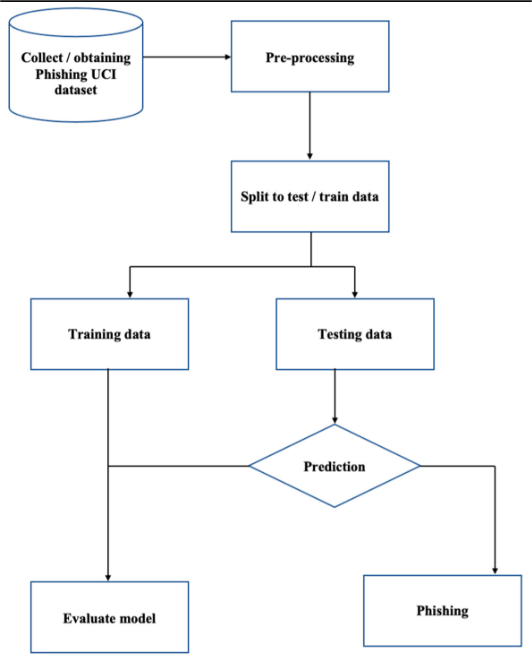
## Data Flow Diagram



# Model training/validation workflow



# User I/O workflow



# Test cases

|  |  |  |  |
| --- | --- | --- | --- |
| Test case | Steps to perform test case | Module | Pass/Fail |
| |  | | --- | | **Accuracy**  **Calculation** |  |  | | --- | |  | | 1. Input a set of labeled phishing and non-phishing domains.  2. Compare system predictions with true labels.  3. Calculate the accuracy of the system. | Accuracy Measurement | pass |
| ROC Curve Generation | 1. Input a set of labeled phishing and non-phishing domains.  2. Generate the ROC curve based on the system's predictions and true labels.  3. Verify if the ROC curve is plotted correctly | ROC Analysis | pass |
| True Positive Rate (TPR) Calculation | 1. Input a set of labeled phishing domains.  2. Calculate the True Positive Rate based on the system's predictions. | ROC Analysis | pass |
| False Positive Rate (FPR) Calculation | 1. Input a set of labeled non-phishing domains.  2. Calculate the False Positive Rate based on the system's predictions. | ROC Analysis | pass |
| AUC Score Calculation | 1. Input a set of labeled phishing and non-phishing domains.  2. Calculate the Area Under the ROC Curve (AUC) score.  3. Verify if the AUC score is above an acceptable threshold. | ROC Analysis | pass |

# Key performance indicators (KPI)

For your phishing domain detection project, Key Performance Indicators (KPIs) are essential to measure the effectiveness and efficiency of your model. Here are some relevant KPIs:

1. **Accuracy**:
   * **Definition**: The proportion of correctly classified domains (both phishing and non-phishing) out of the total domains.
   * **Importance**: Indicates overall performance of the model in classifying domains correctly.
2. **True Positive Rate (TPR) or Recall**:
   * **Definition**: The proportion of actual phishing domains correctly identified by the model.
   * **Importance**: Measures the model's ability to detect phishing domains.
3. **False Positive Rate (FPR)**:
   * **Definition**: The proportion of legitimate domains incorrectly identified as phishing.
   * **Importance**: Indicates how often the model wrongly classifies legitimate domains as phishing.
4. **Precision**:
   * **Definition**: The proportion of correctly identified phishing domains out of all domains classified as phishing.
   * **Importance**: Measures the model's accuracy in predicting phishing domains versus false positives.
5. **F1 Score**:
   * **Definition**: The harmonic mean of precision and recall.
   * **Importance**: Provides a balanced measure of the model's performance, especially when dealing with imbalanced datasets.
6. **Receiver Operating Characteristic (ROC) Curve**:
   * **Definition**: A graphical representation of the true positive rate versus the false positive rate at various thresholds.
   * **Importance**: Helps visualize the trade-offs between sensitivity and specificity and assess the model’s performance across different thresholds.
7. **Area Under the ROC Curve (AUC-ROC)**:
   * **Definition**: The area under the ROC curve.
   * **Importance**: Provides a single scalar value to assess the model’s ability to discriminate between phishing and non-phishing domains. Higher values indicate better performance.
8. **Confusion Matrix Metrics**:
   * **Definitions**: Metrics derived from the confusion matrix such as True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN).
   * **Importance**: Provides detailed insights into how well the model is classifying each category.
9. **Model Robustness**:
   * **Definition**: The model’s performance stability across different datasets or variations in input data.
   * **Importance**: Ensures that the model maintains its effectiveness across various scenarios.

These KPIs will help you gauge the performance, effectiveness, and efficiency of your phishing domain detection system.