**High Level Design (HLD)**

**“PHISHING DOMAIN DETECTION”**

**Revision number:2.0**

**Last date of revision: 10/06/2024**

**Document Version Control**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date Issued** | **Version** | **Description** | **Author** |
| 10/06/2024 | 1 | Initial HLD-V1.0 | SAHIL SAHARMA |
| 10/06/2024 | 2 | Updated Conclusion-V1.2 | SAHIL SHARMA |

***Contents Page No.***

**Document Version Control** 2

**Abstract**  4

1. **Introduction** 5
   1. Why this High-Level Design Document 5
   2. 1.2) Scope

1. **General Description** 6
   1. Problem Statement 6
   2. Objective 6
   3. Approach 6
   4. Tools Used 6-7
2. **Design Details** 8
   1. Process Flow 8
   2. Model Training and Evaluation 9
   3. Error Handling 9
3. **Performance**  10
   1. Reusability 10
   2. Application Compatibility 10
   3. Resource Utilization 10
4. **Key Performance Indicators** 11
5. **Conclusion**  12

# Abstract

Phishing attacks are a pervasive threat in the online world, tricking users into revealing sensitive information through deceptive websites impersonating legitimate entities. This abstract explores the use of automated techniques for phishing domain detection, a crucial step in safeguarding users from such attacks.

* **The Challenge:** Briefly highlight the limitations of traditional methods like blacklists, which can be bypassed by attackers. Emphasize the need for more sophisticated techniques to identify evolving phishing tactics.
* **Machine Learning Approaches:** Introduce the concept of using machine learning algorithms to analyze domain features like URL structure, presence of hyphens, and content characteristics. Briefly mention the effectiveness of these approaches, citing accuracy rates achieved in research.

## 1. Introduction

**1.1 Why this High-Level Design Document?**

The goal of this High-Level Design (HLD) Document is to provide the current project description with the additional depth needed to describe an appropriate model for coding. This paper can serve as a reference guide for how the modules interact at a higher level and is also meant to assist identify conflicts before coding.

**Details on HLD:**

* Present all of the design elements and fully describe them
* explains the user interface that is being used
* explains the software and hardware interfaces
* describes the necessary performance standards
* Include the project's architecture and design elements
* List and explain the non-functional characteristics like those listed below

(Security, Reliability, Maintainability, Portability, Reusability, Application compatibility, Resource utilization, Serviceability)

**1.2 Scope**

The HLD documentation outlines the system's architecture, including the technology architecture, application architecture (layers), database architecture, and application flow. The HLD employs words that should be clear to the system's management, ranging from non-technical to technical.

## 2. General Description

### 2.1 Problem Statement

The burgeoning problem of phishing attacks necessitates the development of robust and automated methods for identifying malicious domains. Phishing websites masquerade as legitimate entities (e.g., banks, social media platforms) to steal sensitive user information like passwords and credit card details. Traditional security measures, such as blacklists, are often ineffective due to:

* **Evolving Tactics:** Phishers constantly adapt their techniques, registering new domains and employing social engineering tricks to bypass blacklisting efforts.
* **Limited Scope:** Blacklists rely on pre-identified malicious domains, failing to detect novel phishing attempts.

**2.2 Objective**

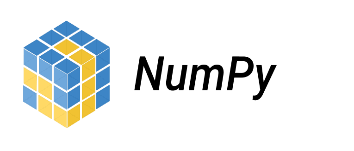
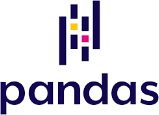
Predicting which entities are fraud and which entities are genuine is the key goal.

#### 2.3 Approach

the traditional machine learning activities, including model building, model testing, feature engineering, data exploration, and data cleaning. applied various machine learning techniques that suited the aforementioned case the best.

#### 2.4 Tools Used

Models are built using the Python programming language and libraries like NumPy, Pandas, and Scikit-learn.



* Seaborn and Matplotlib are used for visualization.

* Deployment is done in local host.

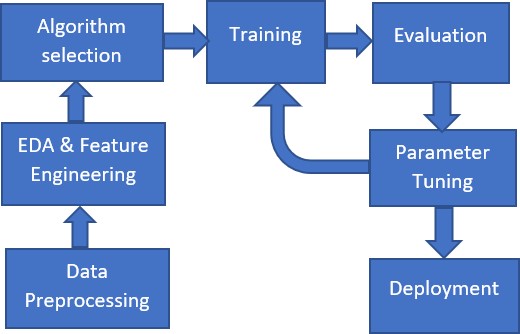
* GitHub is used as version control system.

‘

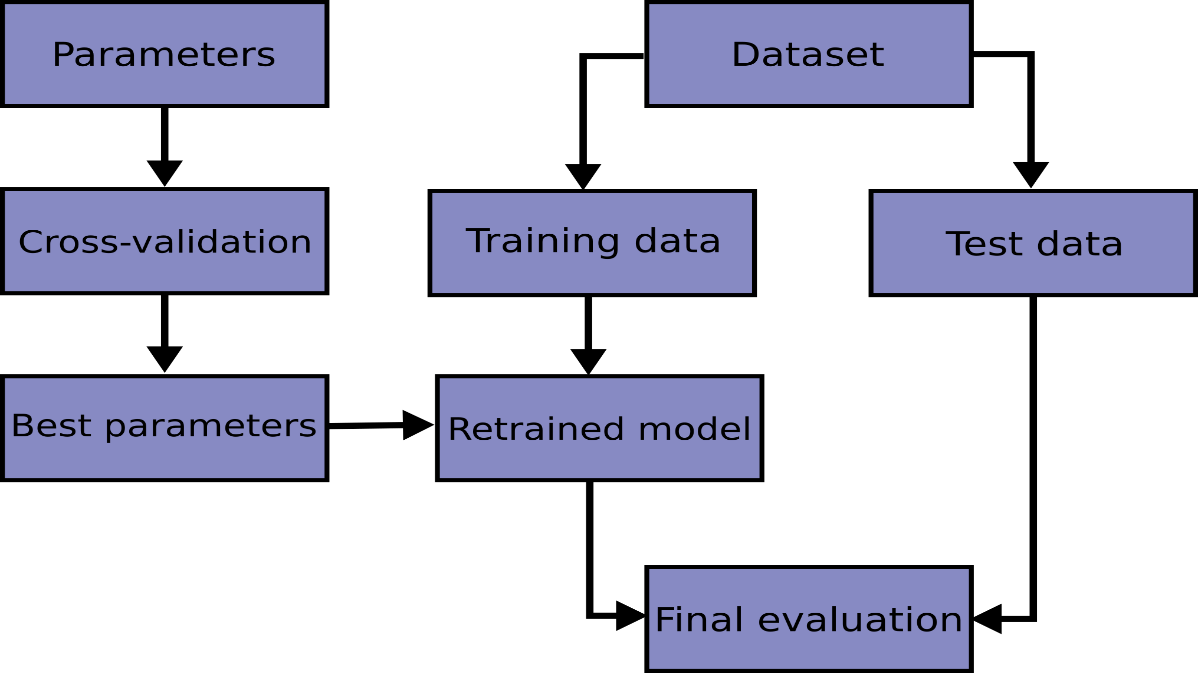
### 3. Design Details

#### 3.1 Process Flow

The following is the procedure for constructing the machine learning model, which is used to categorise edible and poisonous mushrooms.



#### 3.2 Model Training and Evaluation



#### 3.3 Error Handling

Should errors can be encountered. An error will be defined as anything that falls outside the normal intended usage.

### 4. Performance

Phishing domain detection aims for high accuracy, but achieving it perfectly is difficult. Phishing sites mimic real ones and new tactics emerge constantly.

Performance is measured through metrics like precision (avoiding false positives) and recall (catching most phishing attempts). Feature engineering (analyzing URLs, content, historical data) and advanced machine learning algorithms help the system learn and adapt. Regular retraining with new data on identified attacks keeps the system on top of evolving threats.

#### 4.1 Reusability

It should be possible to reuse the written code and the component utilised without any issues.

#### 4.2 Application Compatibility

Python will serve as an interface between the project's many components. Each component will have a specific task to do, and it is Python's responsibility to make sure that the information is transformed correctly.

#### 4.3 Resource Utilization

Any task that needs to be completed will probably consume all of the available computing power.

### 5. Key Performance Indicators

Since achieving perfect accuracy is a challenge, we rely on a set of KPIs to evaluate the effectiveness of a phishing domain detection system:

**Precision:** This KPI measures the percentage of identified phishing domains that are truly malicious.exclamation A high precision indicates the system effectively avoids flagging legitimate websites as phishing attempts (low false positive rate).

**Recall:** This KPI measures the percentage of actual phishing domains that the system successfully identifies.exclamation A high recall ensures the system catches most phishing attempts and minimizes the risk of users encountering them (low false negative rate).

**F1-Score:** This metric combines precision and recall into a single score, providing a balanced view of the system's performance.expand\_more A high F1-Score indicates the system achieves a good balance between avoiding false positives and catching true positives.expand\_more

**Detection Time:** This measures the time it takes for the system to identify a new phishing domain. A faster detection time minimizes the window of opportunity for attackers.expand\_more

**False Positive Rate (FPR):** This is the direct opposite of precision and represents the percentage of legitimate domains incorrectly flagged as phishing. A low FPR is desirable.exclamation

**False Negative Rate (FNR):** This is the opposite of recall and represents the percentage of phishing domains the system misses. A low FNR is crucial.

### 6. Conclusion

Phishing attacks pose a significant threat to online security, tricking users into surrendering sensitive information. Phishing domain detection systems play a critical role in safeguarding users by identifying and blocking malicious websites. While achieving perfect accuracy is a challenge due to the deceptive nature and constant evolution of phishing tactics, significant strides can be made.

By employing advanced machine learning algorithms, meticulously crafted features, and continuous learning strategies, phishing domain detection systems can achieve high performance. Metrics like precision, recall, and F1-score provide valuable insights into the system's effectiveness.

The fight against phishing is an ongoing one. As phishers develop new techniques, so too must our detection systems adapt. Through continuous research, improvement, and collaboration, we can build robust defenses that significantly reduce the risk of users falling victim to phishing attacks and create a safer online environment.