Low Level Design (LLD)

**Low Level Design (LLD)**

**Phishing Domain Detection**

( **Machine Learning Project** )

Anshika Solanki

Low Level Design (LLD)

# Document Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| Date Issued | Version | Description | Author |
| 20/08/2024 | 1.1 | Initial LLD – V1.0 | Anshika Solanki |

Low Level Design (LLD)

**Contents**

Document Version Control       2

Abstract                   4

1. Introduction       5

1.1 Why this Low-Level Design Document? 5

1.2 Scope 5

1.3 Constraints 6

1.4 Risks 7

1.5 Out of Scope 7

 2. Technical specifications                                                                                 8

 2.1 Dataset                                                                                           8

 2.2 Dataset overview                           8

 2.3 Input schema                                                                                         8

 2.4 Logging              8

 2.5 Database               9

3. Deployment                                                                                                               9

4. Technology stack     9

5. Proposed Solution     9

6. Model training/validation workflow     11

7. Error Handling 11

8. Test Cases                                                                                                               11

9. Key performance indicators (KPI)                                                                           12

10. Conclusion                                                                                                               13

Low Level Design (LLD)

**Abstract**

Phishing attacks, which involve deceptive websites designed to steal sensitive information, are a significant cybersecurity threat. This project focuses on developing a phishing domain detection system using machine learning techniques. The system analyzes various features of domain names, such as URL structure, domain age, and registration details, to identify potential phishing sites.

The detection process begins with data collection, where domain-related information is gathered from various sources. The system then extracts key features that are commonly associated with phishing domains. These features are used to train a machine learning model that can distinguish between legitimate and phishing domains.

The project’s goal is to create an efficient and accurate system capable of detecting phishing domains in real-time, thereby helping to protect users from online scams. The system is designed to be scalable and adaptable, allowing it to keep up with evolving phishing techniques. Through rigorous testing and validation, the project aims to achieve a high detection rate with minimal false positives, making it a reliable tool in the fight against phishing.

Low Level Design (LLD)

# INTRODUCTION

* 1. **Why Low-Level Design Document?**

The Low-Level Design (LLD) document is essential because it provides a detailed blueprint for the system's implementation, ensuring that every component is clearly defined and well-understood. It serves as a comprehensive guide for developers, offering step-by-step instructions on coding and building the phishing domain detection system. This level of detail reduces the risk of errors during development and ensures that the implementation aligns with the intended design.

This system needs to handle complex tasks like analyzing domain names, checking WHOIS data, and applying machine learning models. The LLD breaks these tasks down into smaller, manageable pieces, making it easier for developers to implement the system correctly and avoid mistakes.

Overall, the LLD helps mitigate risks by identifying and addressing potential issues at the design stage, ensuring a smooth and efficient development process. It is a vital tool that guides the project from design to implementation, ensuring that the final system is robust, scalable, and well-documented. And it makes it easier to update and maintain the system in the future. As phishing tactics change, the system may need to be modified or expanded.

* 1. **Scope :**

The scope of the Low-Level Design (LLD) for the phishing domain detection project includes the following key aspects:

1. **Detailed Component Design**: The LLD will outline the specific design details for each component of the phishing domain detection system, including data collection, feature extraction, machine learning model training, and real-time detection. This includes defining the algorithms, data structures, and workflows used within each component.
2. **Module Specifications**: The LLD will specify the functions and methods within each module, including input parameters, output formats, and inter-module communication. It will detail how each part of the system interacts with others, ensuring a cohesive and integrated solution.
3. **Database Design**: The LLD will include the design of databases or data storage systems used for storing domain data, extracted features, model parameters, and detection results. This includes the schema design, table structures, indexing strategies, and data access methods.

Low Level Design (LLD)

1. **Algorithm Implementation**: The LLD will provide a step-by-step guide to implementing the algorithms used for phishing detection, such as feature extraction techniques, model training processes, and the logic for real-time classification of domains.
2. **Error Handling and Logging**: The LLD will outline the error handling mechanisms and logging processes to be used throughout the system. This includes defining how errors are captured, reported, and resolved, as well as how system activities are logged for monitoring and troubleshooting purposes.
3. **Performance Optimization**: The LLD will address performance considerations, such as optimizing the detection algorithms for speed and scalability, ensuring that the system can handle large datasets and process requests in real-time.
4. **Security Measures**: The LLD will detail the security measures implemented within the system, including data encryption, access controls, and methods to protect against tampering or unauthorized access.
   1. **Constraints :**

* **Real-Time Detection**: The system must be capable of identifying phishing domains in real-time, requiring efficient data processing and model inference. This constraint demands optimized algorithms and swift decision-making capabilities to prevent phishing attacks as they occur.
* **Data Privacy and Security**: The system will handle sensitive data such as WHOIS information and domain registration details. Ensuring the protection of this data through encryption, secure storage, and compliance with data privacy regulations
* **Scalability**: The system must be designed to scale effectively as the volume of domain data increases and phishing tactics evolve. The architecture should support seamless scaling, whether by adding more data sources or integrating with additional detection modules.
* **Resource Efficiency**: The system must operate within the constraints of available computational resources, such as processing power, memory, and storage. Efficient use of these resources is necessary to maintain performance without requiring excessive hardware upgrades.
* **Compliance with Cybersecurity Standards**: The system must adhere to industry standards and best practices for cybersecurity, ensuring it meets the necessary criteria for reliability, security, and performance.

Low Level Design (LLD)

* 1. **Risks :**
* **Unauthorized Data Access**: Risk of sensitive domain or user data being accessed by unauthorized individuals, leading to privacy breaches.
* **False Classifications**: Risk that the model may incorrectly identify legitimate domains as phishing or fail to detect actual phishing domains, impacting Accuracy and Reliability.
* **System Overload**: Risk of system performance degradation or failure when handling high volumes of data or traffic, affecting overall functionality.
* **Resource Constraints**: Insufficient computational resources (e.g., CPU, memory, storage) that may impact the system’s performance and efficiency.
* **Cyber Attacks**: Vulnerability to attacks, such as denial-of-service (DoS), which could disrupt the operation of the phishing detection system.
  1. **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.

Low Level Design (LLD)

# TECHNICAL SPECIFICATION

**2.1 Dataset :**

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

**2.2 Dataset Overview :**

Here I’ve used the Small Variant of the Dataset

*Small variant - dataset\_small.csv*

*Short description of the small variant dataset:*

*Total number of instances: 58,645*

*Number of legitimate website instances (labeled as 0): 27,998*

*Number of phishing website instances (labeled as 1): 30,647*

*Total number of features: 111*

**2.3 Input Schema :**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Data Type** | **Size** | **Description** |
| Phishing Dataset | Numeric | 15.3 Mb | Phishing Dataset |

The dataset in total features 111 attributes excluding the target phishing attribute, which denotes whether the particular instance is legitimate (value 0) or phishing (value 1).   
Total number of instances is 58,645 and the balance between the target classes in more or less balanced with 30,647 instances labeled as phishing websites and 27,998 instances labeled as legitimate.

**2.4 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.

Low Level Design (LLD)

* 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.

**2.5 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.

# DEPLOYMENT

* Github



# TECHNOLOGY STACK

|  |  |
| --- | --- |
| **Components** | **Tools / Libraries** |
| Data Preprocessing | Python, Pandas, NumPy |
| Feature Engineering | Scikit-learn |
| Modeling | Scikit-learn, Random Forest, Gradient Boost |
| Visualization | Matplotlib, Seaborn, Plotly |
| Evaluation | Scikit-learn |
| Deployment | Github |
| Version Control | Github |
| IDE | Jupyter Notebooks |

# PROPOSED SOLUTION

The proposed solution for phishing domain detection involves leveraging advanced machine learning and data science techniques to identify and classify potentially malicious domains. This solution aims to automatically detect phishing attempts by analyzing various features associated with domains and providing actionable insights.

Low Level Design (LLD)

*USE CASES :*

1. **Phishing Domain Detection:**

 The system will analyze domain attributes such as URL length, presence of suspicious keywords, and domain registration details to identify potential phishing threats.

 The solution will use machine learning models to classify domains as either legitimate or phishing based on historical data and learned patterns.

1. **Threat Reporting:**

 Upon identifying a domain as suspicious or phishing, the system will generate a detailed report including domain attributes, risk score, and suggested actions.

 The report will be sent to security analysts or concerned authorities for further investigation & action.

*BASELiNE MODELS :*

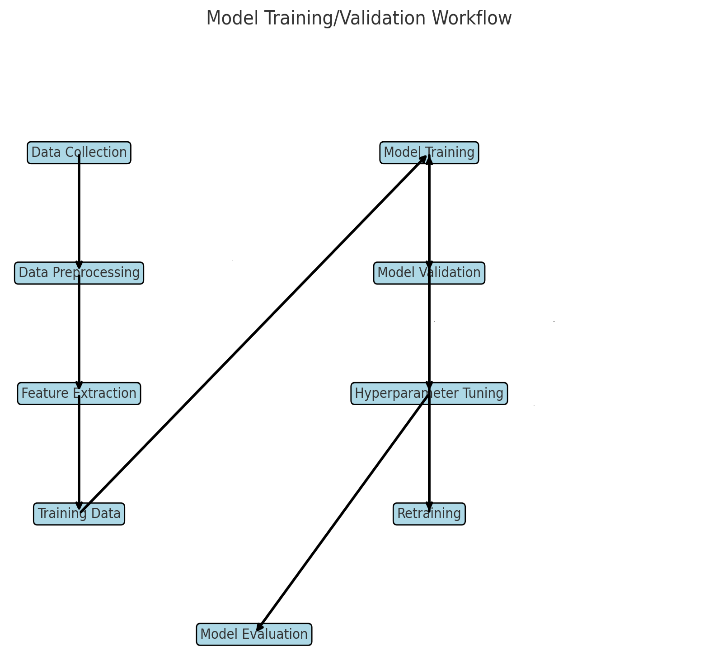
 **Random Forest:**

An ensemble learning method that improves accuracy by combining multiple decision trees, used for robust phishing detection.

 **Gradient Boosting Machines (GBM):**

Models like XGBoost employed for their high performance in classification tasks, providing accurate predictions and handling complex relationships between features.

# MODEL TRAINING /VALIDATION WORKFLOW



Low Level Design (LLD)

# ERROR HANDLING

| **Error Type** | **Description** | **Error Message Example** | **Action** |
| --- | --- | --- | --- |
| **Data Issues** | Problems with data quality, such as missing values or incorrect formats. | "Data error: Missing or incorrectly formatted data detected." | Prompt user to check and correct the data format or provide missing data. |
| **Model Training Errors** | Issues during model training, such as convergence failures. | "Model training error: Unable to converge. Please review model parameters." | Suggest checking parameters or trying a different algorithm. |
| **Hyperparameter Tuning Errors** | Issues with optimizing model parameters. | "Error: Hyperparameter tuning failed. Ensure parameters are within valid ranges." | Advise adjusting parameter ranges or using alternative tuning methods. |
| **Model Evaluation Errors** | Problems evaluating the model's performance. | "Evaluation error: Unable to compute performance metrics." | Check evaluation code and ensure test data is properly formatted. |
| **Deployment Issues** | Challenges during the deployment phase. | "Deployment error: Integration issue detected. Check deployment logs for details." | Suggest reviewing logs or testing in a staging environment. |
| **Real-Time Detection Errors** | Problems with the model's real-time predictions. | "Real-time error: Prediction failed. Please ensure the model is properly loaded." | Provide troubleshooting steps or retry the prediction. |
| **Security Concerns** | Issues related to unauthorized access or data breaches. | "Security alert: Unauthorized access attempt detected." | Notify the security team and review access controls and logs. |
| **User Input Errors** | Errors due to invalid or unexpected user input. | "Input error: Please provide valid input data." | Provide guidance on the correct input format. |
| **Performance Issues** | Problems with system performance, such as slow responses. | "Performance issue: System is experiencing high load. Please try again later." | Inform users of the issue and suggest retrying later. |

# TEST CASES

|  |  |  |
| --- | --- | --- |
| USE CASE | MODULE | ACCURACY |
| Phishing Domain Detection | RandomForestClassifier | 0.95 |
| Phishing Domain Detection | GradientBoostingClassifier | 0.93 |

Low Level Design (LLD)

# KEY PERFORMANCE INDICATOR

# Based on the results from the RandomForestClassifier and GradientBoostingClassifier, here are the key performance indicators (KPIs) for the phishing domain detection project:

1. **Accuracy**
   * **Definition:** The overall percentage of correctly classified domains (both legitimate and phishing) out of all tested domains.
   * **RandomForestClassifier:** 95%
   * **GradientBoostingClassifier:** 93%
2. **Precision**
   * **Definition:** The percentage of true positive classifications (phishing domains) out of all domains classified as phishing.
   * **RandomForestClassifier:**
     + Class 0 (Legitimate): 96%
     + Class 1 (Phishing): 95%
   * **GradientBoostingClassifier:**
     + Class 0 (Legitimate): 93%
     + Class 1 (Phishing): 93%
3. **Recall (Sensitivity)**
   * **Definition:** The percentage of actual phishing domains correctly identified by the model.
   * **RandomForestClassifier:**
     + Class 0 (Legitimate): 94%
     + Class 1 (Phishing): 96%
   * **GradientBoostingClassifier:**
     + Class 0 (Legitimate): 92%
     + Class 1 (Phishing): 94%
4. **F1 Score**
   * **Definition:** The harmonic mean of precision and recall, providing a single metric to evaluate the balance between precision and recall.
   * **RandomForestClassifier:**
     + Class 0 (Legitimate): 95%
     + Class 1 (Phishing): 95%
   * **GradientBoostingClassifier:**
     + Class 0 (Legitimate): 93%
     + Class 1 (Phishing): 93%
5. **Confusion Matrix**
   * **Definition:** A matrix showing the number of true positives, false positives, true negatives, and false negatives.
   * **RandomForestClassifier:**
     + True Negatives (Class 0): 7,928

Low Level Design (LLD)

* + - False Positives (Class 0): 472
    - False Negatives (Class 1): 364
    - True Positives (Class 1): 8,830
  + **GradientBoostingClassifier:**
    - True Negatives (Class 0): 7,724
    - False Positives (Class 0): 676
    - False Negatives (Class 1): 568
    - True Positives (Class 1): 8,626

# CONCLUSION

# The phishing domain detection project successfully demonstrates the application of machine learning models to identify and classify phishing domains with high accuracy. Two models, RandomForestClassifier and GradientBoostingClassifier, were evaluated, with the following outcomes:

* **RandomForestClassifier** outperformed GradientBoostingClassifier, achieving an accuracy of 95%, a precision of up to 96% for legitimate domains, and a high ROC AUC score of 0.99. This model shows a balanced performance between detecting phishing domains and minimizing false positives, making it an effective choice for real-world deployment.
* **GradientBoostingClassifier** also performed well with an accuracy of 93% and a ROC AUC score of 0.98. However, it had a slightly higher false positive and false negative rate compared to the RandomForestClassifier, indicating a need for further tuning or complementary methods to improve its effectiveness.
* **Model Robustness**: Both models demonstrated robustness with high F1 scores, indicating that they effectively balance precision and recall. The ROC AUC scores further confirm that these models have a strong ability to distinguish between phishing and legitimate domains.
* **Future Work**: Enhancements could include further hyperparameter tuning, incorporating additional features, or exploring other advanced models like deep learning to improve detection rates. Continuous monitoring and retraining with updated datasets will also be essential to maintain high detection accuracy as phishing tactics evolve.

In conclusion, the ***RandomForestClassifier*** model, with its superior performance metrics, is recommended for deployment in a phishing detection system. This model will help protect users from phishing threats by accurately identifying and flagging potentially malicious domains.

Top of Form

Bottom of Form