**Low-Level Design Documentation (LLD)**



**Project On:**

**Title: Phishing Domain Detection**

**By: - Dharavath Ramdas**

**(iNeuron.ai)**

**Date: - 12/03/2023**

**Contents**

1. **Abstract**
2. **Introduction**
   1. **Importance of Low-Level Design (LLD)?**
   2. **Scope of (LLD)**
3. **Architecture**
   1. **Architecture Design**
   2. **Data Requirement.**
   3. **Data Collection.**
   4. **About the dataset.**
   5. **Data Description**
   6. **Tools / Software Used.**
   7. **Importing data into databases**
   8. **Exporting data from the databases**
   9. **Data pre-processing**
   10. **Modeling**
   11. **UI integration**
   12. **Data from user**
   13. **Data validation**
   14. **Rendering the results**
4. **Deployment**
5. **Abstract.**

Phishing is a type of fraud in which an attacker impersonates a reputable company or person in order to get sensitive information such as login credentials or account information via email or other communication channels. Phishing is popular among attackers because it is easier to persuade someone to click a malicious link that appears to be authentic than it is to break through a computer's protection measures.

Phishing stands for a fraudulent process, where an attacker tries to obtain sensitive information from the victim. Usually, these kinds of attacks are done via emails, text messages, or websites. Phishing websites, which are nowadays in a considerable rise, have the same look as legitimate sites. However, their backend is designed to collect sensitive information that is inputted by the victim. Discovering and detecting phishing websites has recently also gained the machine learning community’s attention, which has built the models and performed classifications of phishing websites. This paper presents two dataset variations that consist of 58,645 and 88,647 websites labeled as legitimate or phishing and allow the researchers to train their classification models, build phishing detection systems, and mining association rules.

The mail goal is to predict whether the domains are real or malicious

1. **Introduction.**

2.1. What is a Low-Level design document?

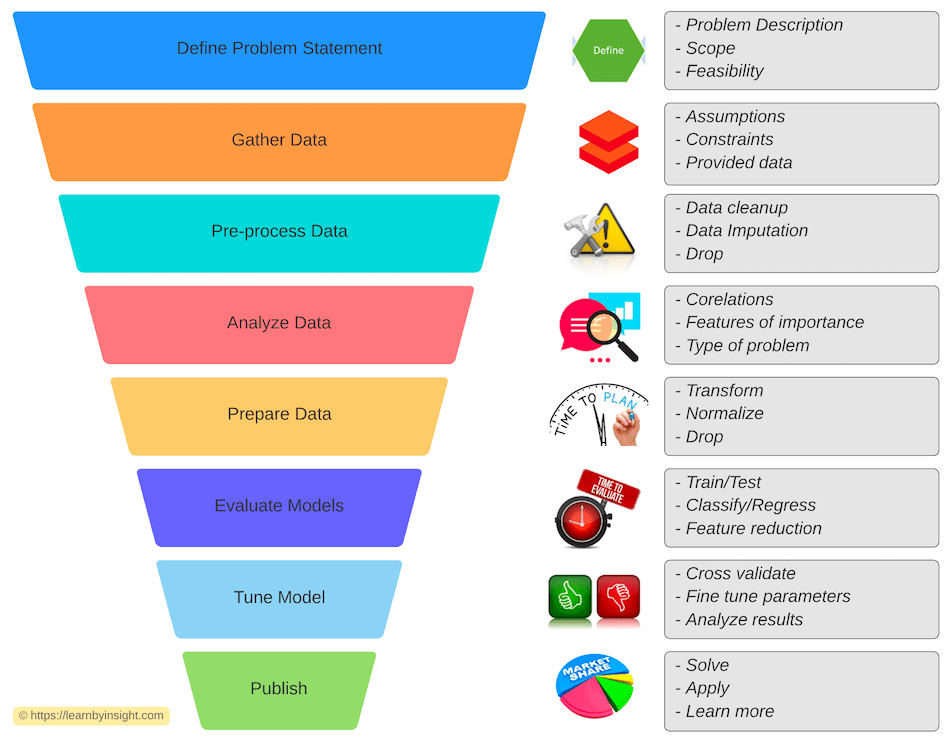
The goal of LLD or a low-level design document (LLDD) is to give the internal

logical design of the actual program code for Food Recommendation System. LLD describes the class diagrams with the methods and relations between classes and program specs. It describes the modules so that the programmer can directly code the program from the document.

2.2. Scope of (LLD)

Low-level design (LLD) is a component-level design process that follows a step-by-step refinement process. This process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work

1. **Architecture.**



**3.1 ARCHITECTURE DESIGN**

This project is completely based on the life cycle of machine learning, where we will be predicting the insurance premium. The tools used in this project are Python, Pandas, NumPy, Matplotlib, Seaborn, Scikit learn,

For the Version Control system Git was used and for continuous deployment **AWS (amazon web service)** is used and for web app **Microsoft Azure** is used.

**3.2 Data Requirement.**

Whenever we are working on any project the data is completely dependent on the requirement of the problem statement. For this project the problem statement was to create a Hyper tuned Classification machine learning model which can predict the whether the domains are real or malicious.

**3.3 Data Collection.**

Dataset link: [https://data.mendeley.com/datasets/72ptz43s9v/1](https://data.mendeley.com/datasets/72ptz43s9v/1%20)

**3.4 About the dataset.**

These data consist of a collection of legitimate as well as phishing website instances. Each website is represented by the set of features which denote, whether website is legitimate or not. Data can serve as an input for machine learning process.

In this repository the two variants of the Phishing Dataset are presented.

Full variant - dataset\_full.csv

Short description of the full variant dataset:

Total number of instances: 88,647

Number of legitimate website instances (labeled as 0): 58,000

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

Total number of features: 111

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

**3.5 Data Description.**

Below are the various attributes for the Insurance Phishing Domain Detection Dataset. Please go through the dataset\_full.csv for better clarity of the attributes when reading this document.

• attributes based on the whole URL properties presented in [Table 1](https://www.sciencedirect.com/science/article/pii/S2352340920313202" \l "tbl0001),

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] |

• attributes based on the domain properties presented in [Table 2](https://www.sciencedirect.com/science/article/pii/S2352340920313202" \l "tbl0002),

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] |

• attributes based on the URL directory properties presented in [Table 3](https://www.sciencedirect.com/science/article/pii/S2352340920313202" \l "tbl0003),

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 |  |

• attributes based on the URL file properties presented in [Table 4](https://www.sciencedirect.com/science/article/pii/S2352340920313202" \l "tbl0004),

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 |  |

•attributes based on the URL parameter properties presented in [Table 5](https://www.sciencedirect.com/science/article/pii/S2352340920313202" \l "tbl0005), and

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 |  |

•attributes based on the URL resolving data and external metrics presented in [Table 6](https://www.sciencedirect.com/science/article/pii/S2352340920313202" \l "tbl0006).

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]** |

The first group is based on the values of the attributes on the whole URL string, while the values of the following four groups are based on the particular sub-strings, as presented in [Figure 1](https://www.sciencedirect.com/science/article/pii/S2352340920313202" \l "fig0001). The last group attributes are based on the URL resolve metrics as well as on the external services such as Google

search index.

Fig. 1

Fig. 1. Separation of the whole URL string into sub-strings.

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). We prepared two variations of the dataset, the one where the 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. The second variant of the dataset is comprised of 88,647 instances with 30,647 instances labeled as phishing and 58,000 instances labeled as legitimate, the purpose of which is to mimic the real-world situation where there are more legitimate websites present. The distribution between the classes of both dataset variants is presented in [Figure 2](https://www.sciencedirect.com/science/article/pii/S2352340920313202" \l "fig0002).

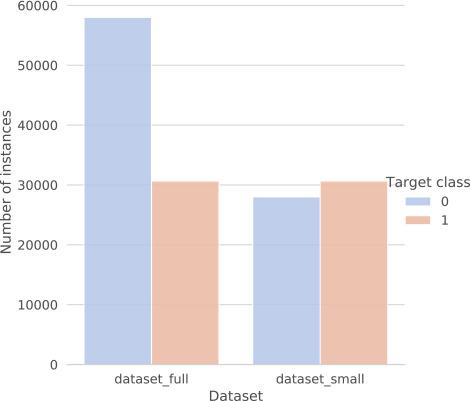


Fig. 2. The distribution between classes for both dataset variations. The *dataset\_full* denotes the larger dataset, while the *dataset\_small* denotes the smaller dataset variation. The target class *0* denotes legitimate websites while the target class *1* denotes the phishing websites.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |

**3.6 Tools / Software Used:-**

* Python version used for this project 3.8 (This may get updated and some features might not be available in new version.)
* Python libraries such as NumPy, pandas, flask, Jinja2, HTML, matplotlib, PyMongo, seaborn and scikit-learn (Used for implementation of machine learning algorithms.)
* Jupiter Visual studio code is used as an IDE for writing the code.
* Github is used as the version control system.
* AZURE is used for deployment.

**3.7 Importing data into the databases.**

MongoDB was used for loading the dataset using Pandas Library was used for training and making the machine-learning model.

**3.8 Exporting data to the database**

The data has been dumped to the MongoDB database..

**3.9 Data Preprocessing**

Have taken the dataset\_full.csv file as my dataset.

* All the necessary libraries were imported first such as NumPy, pandas, flask, Jinja2, HTML, matplotlib, PyMongo, seaborn and scikit-learn (Used for implementation of machine learning algorithms.)
* Checking the basic profile of the dataset. To get a better understanding of the dataset.
  + Using Info method
  + Using Describe method
  + Checking for unique values of each column.
* Checking for null values, there are no null values present in our dataset.
* After performing all the above steps, the dataset is ready and can be processed into the stage of modelling.

**3.10 Modeling**

* After this the data was split into 2 sets X and y. X contains all the columns except the target column in our case (expenses), and y contains only the Target column.
* Using train test split we first split the dataset into X\_train, X\_test, y\_train, and y\_test.
* Standard scaling has been used to bring the data on the same scale
* The following libraries were imported to create classification models.
  + For prediction, xgboost classifier has been chosen as a final algorithm to create the model

**3.11 UI Integration**

Apache Airflow can be used to monitor the model and predict the new batch dataset. A flask webapp has been created to get the Phishing Domain Detection.

WebApp link: http://project102-env.eba-2btzpcth.us-east-1.elasticbeanstalk.com/

**3.12 Data from the user**

User can give the required input the URL of any website as a result in Flask webapp is used to predict whether the url/domains are real or malicious. Data from the user is retrieved using the batch file and using Apache Airflow our machine learning model to give the predicted result.

**3.13 Data Validation**

The data which is entered by the user is validated by the data\_validation.py file which is built using inside the components folder under insurance and then this data is transformed using data\_transformation.py under the same path and finally transferred to our model.

**3.14 Rendering the result**

The result for the predictions can be obtained in the Flask webapp and also result for our model can be seen in the prediction file generated after running the batch prediction in Apache Airflow.

**4. Deployment**

1. This model is deployed on AWS Ec2 instances. The following are the steps to deploy the model on the AWS platform:

* Create an ECR
* Create S3 bucket
* Create an AWS account
* Create an EC2 instance
* Edit security group
* Connect to an EC2 instance
* Install Docker
* Add the runner in the GitHub
* Add all the secret keys in the GitHub
* In the GitHub actions, run the continuous delivery and deployment workflow once after starting the runner in the EC2 instance

1. A web app has been created and deployed using Microsoft Azure