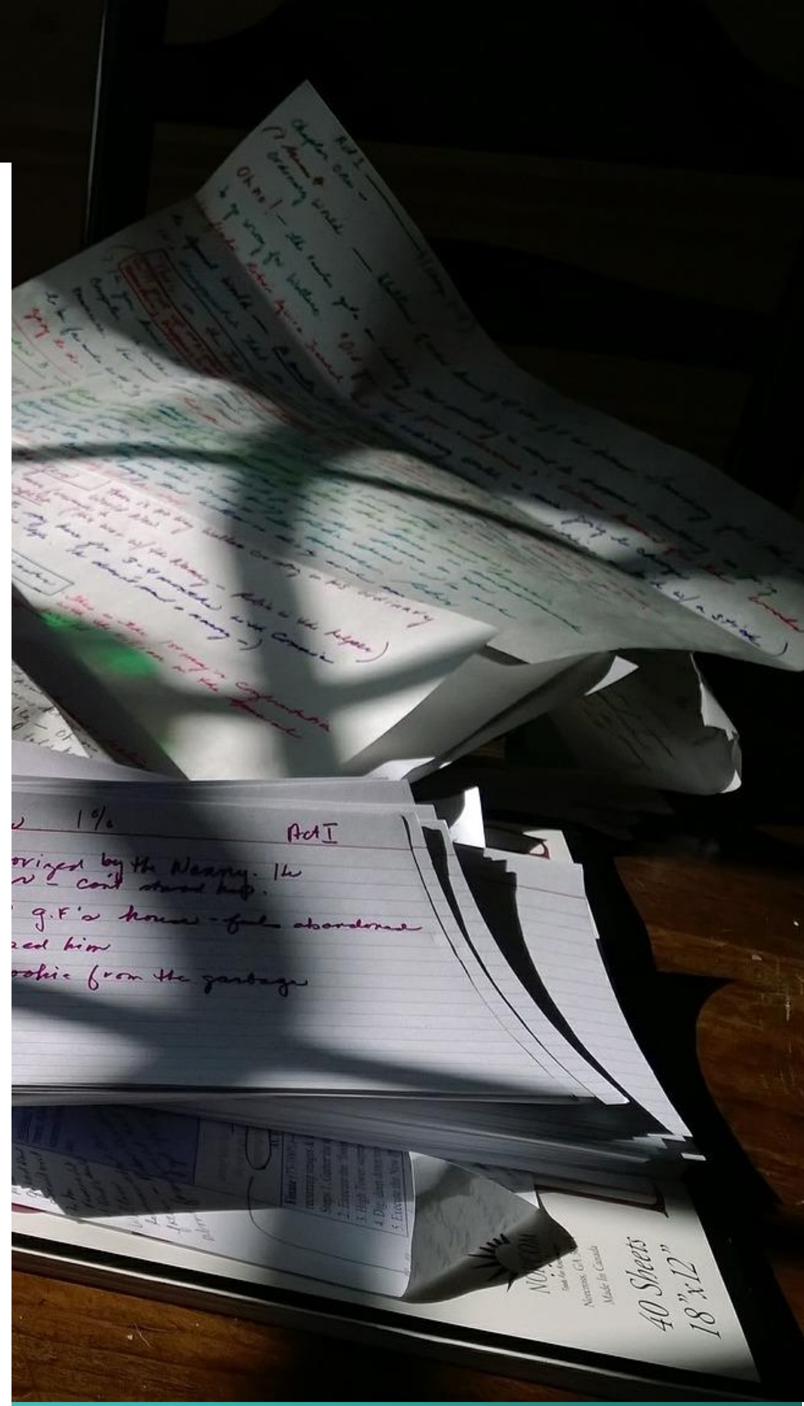
PHISHING DOMAIN DETECTION

Low Level Design (LLD)



NOVEMBER 4

iNeuron.ai | PW Skills
Authored by: Suyash Waykar

PW SKILLS | iNeuron

Abstract

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.

The main goal of the project is to predict if URL is phishing or not.

Why LLD?

The purpose of this document is to present a detailed description of the Phishing Domain Detection Model. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which URL can be detected as phishing. This document is intended for both the stakeholders and the developers of the system and will be proposed to the higher management for its approval.

Phishing is a type of fraud in which an attacker impersonates a reputable company or person 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. As said before the main objective of the project is to predict if a URL is phishing URL or not.

This project shall be delivered in two phases:
Phase 1: All the functionalities with PyPi packages.
Phase2: Integration of UI to all the functionalities.

Scope -

This software system will be a Web application. This system will be designed to detect phishing websites and prevent cyber-attacks and improve cyber security.

Constraints -

For the phishing websites, only the ones from the Phish Tank registry were included, which are verified from multiple users. For the legitimate websites, we included the websites from publicly available, community labelled and organized lists [1], and from the Alexa top ranking websites.

Out of Scope -

Delineate specific activities, capabilities, and items that are out of scope for the project.

Technical Specifications

Dataset -

Data were acquired through the publicly available lists of phishing and legitimate websites, from which the features presented in the datasets were extracted.
Data source: https://doi.org/10.17632/72ptz43s9v.1
Data format Raw: csv file

Value of the Data -

- This data consists of a collection of legitimate, as well as phishing website instances. Each website is represented by a set of features that denote whether the website is legitimate or not. Data can serve as input for the machine learning process.
- Machine learning and data mining researchers can benefit from these datasets, while also computer security researchers and practitioners. Computer security

enthusiasts can find these datasets interesting for building firewalls, intelligent ad blockers, and malware detection systems.

- This dataset can help researchers and practitioners easily build classification models in systems preventing phishing attacks since the presented datasets feature the attributes which can be easily extracted.

Finally, the provided datasets could also be used as a performance benchmark for developing state-of-the-art machine learning methods for the task of phishing websites classification.

Data Description -

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: The data is comprised of the features extracted from the collections of websites addresses. The data in total consists of 111 features, 96 of which are extracted from the website address itself, while the remaining 15 features were extracted using custom Python code.

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]

Detecting Phishing URL -

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 sub-strings, as presented in Figure. The last group attributes are based on the URL resolve metrics as well as on the external services such as Google search index.

Logging -

We should be able to log every activity done by the user.

- The System identifies at what step logging is required.
- The System should be able to log every system flow.
- Developers can choose logging methods. You can choose database logging/ File logging as well.
- System should not be hung even after using so many loggings. Logging just because we can easily debug issues, so logging is mandatory to do.

Database -

System needs to store every request in the database, and we need to store it in such a way that it is easy to retrain the model as well.

1. The User chooses the disease.
2. The User gives required information.
3. The system stores every data given by the user or received on request in the database. Database you can choose your own choice (Cassandra).

Technology Stack

Front End	**HTML/CSS/JS/React**
Backend	**Python Django**
Database	**Cassandra**
Deployment	**AWS**

Proposed Solution

refer: https://www.sciencedirect.com/science/article/pii/S2352340920313202

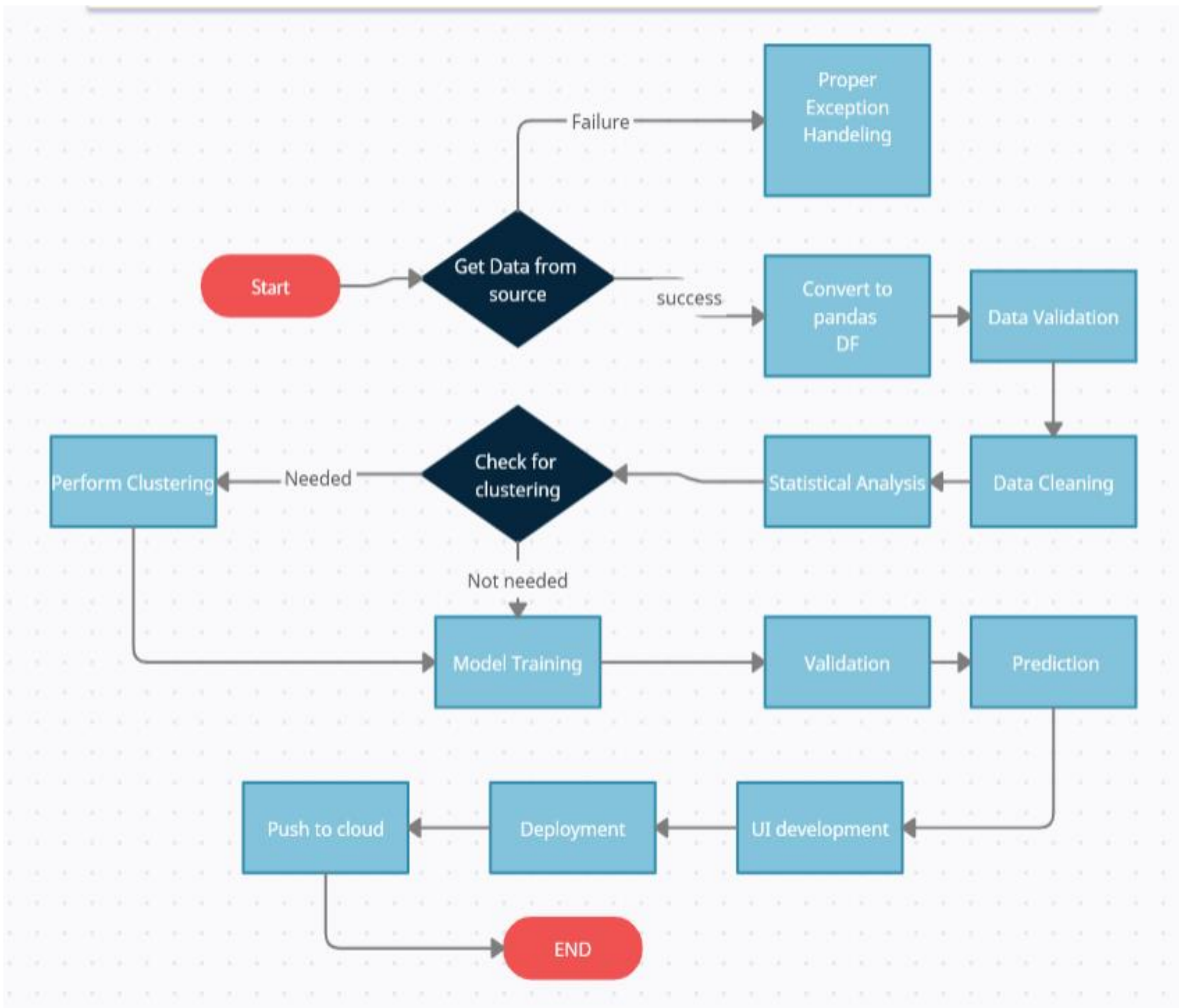
The classical machine learning tasks like Data Exploration, Data Cleaning, Feature Engineering, Model Building and Model Testing. Try out different machine learning algorithms that's best fit for the above case.

For Feature Engineering show:

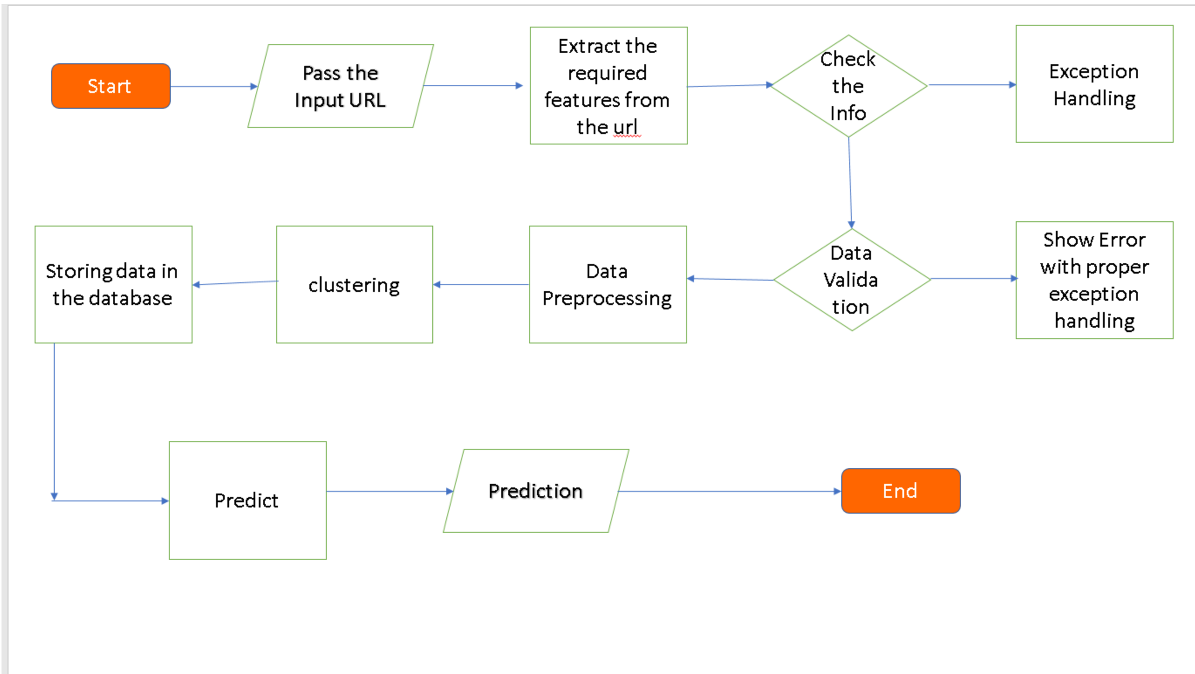
1. URL-Based Features
2. Domain-Based Features
3. Page-Based Features
4. Content-Based Features

Baseline Model: Logistic Regression since this is a classification problem.
Actual model: XG Boast

Model Training/Validation Workflow



User I/O Workflow



Key Performance API

- Detecting malicious websites will help in preventing cyber-attacks.
- Comparison of accuracy of model prediction and real time data prediction.

THANKS!