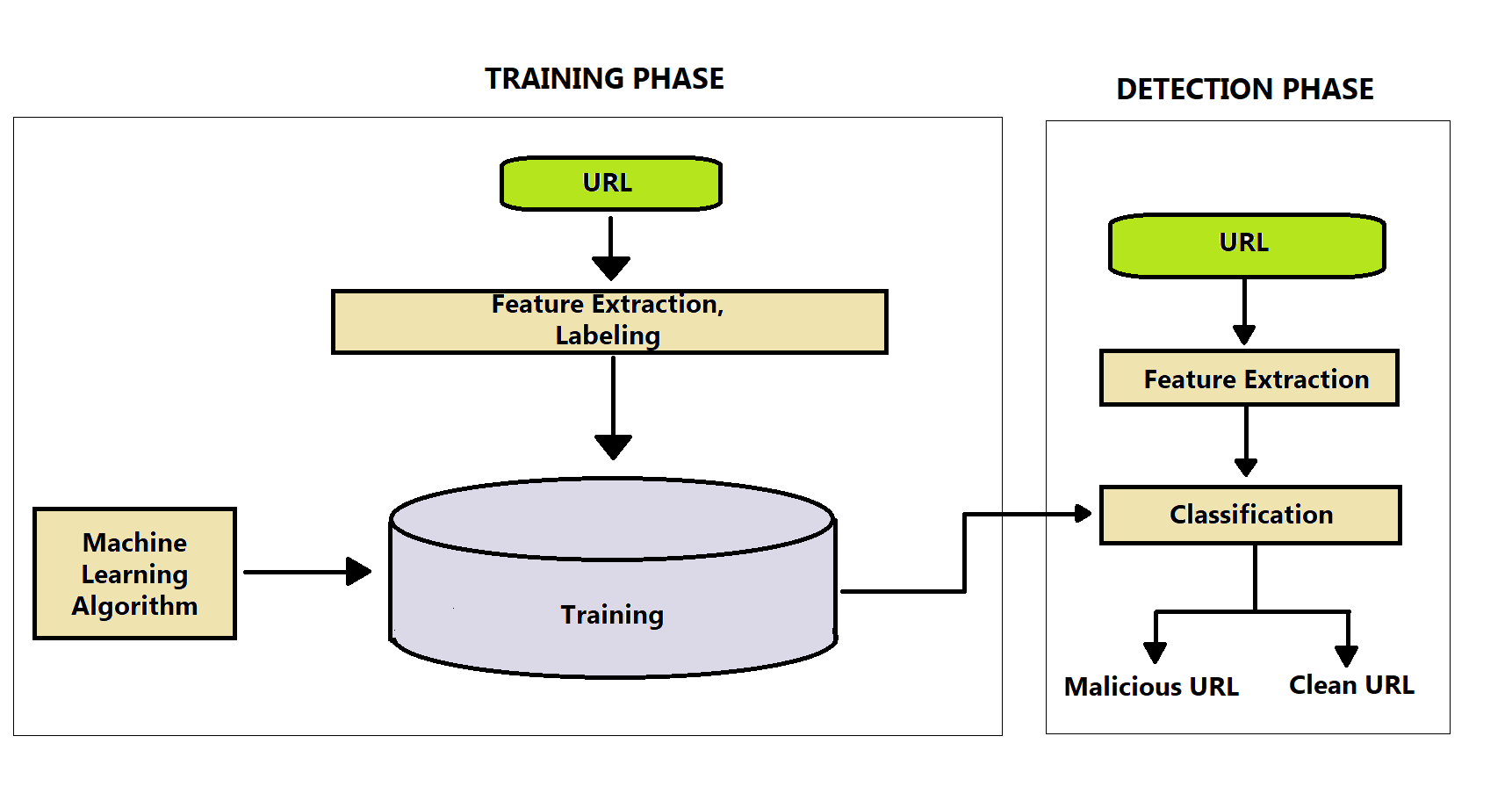
System Architecture

Here we are going to see the flow which we follow to predict the number of malicious and clean URLs. The general thinking behind used machine learning is to make the algorithm which can learn by itself the best parameters for data to make good predictions.

In fig \*\*\*\*, The malicious URL detection system using machine learning contains two phases: training and detections.

Training Phase: It is very necessary for the detection of malicious URLs is to collect both malicious and clean URLs. Then collected both URLs into single labelled and proceed it to attribute extraction. These attributes will be the best basis for detecting both clean URLs and malicious URLs. Further, this dataset will be divided into 2 subsets: training data which is used for training machine learning algorithms, and testing data which is used for testing process. If the classification performance of the model is good and achieve high accuracy, then it will be used in the detection phase.

Detection Phase: It performed on each input URL. Firstly, URL will go through attribute extraction process. Next, these attributes processed as input to the classifier to classify whether the URL is clean or malicious.



URL attribute extraction and selection

Listed some main attribute group for URL detections as follow:

1. Lexical Features: it includes URL length, main domain length, maximum token domain length, path average length, average token length in domain.
2. Host-based Features: it extracted from host characteristics of the URLs. These attributes indicate the location and identity of malicious servers, the degree of impact of several host-based features that contribute in URL’s malicious level.
3. Content-based Features: it acquired when a whole web page is installed. Since a lot of information needs to be extracted and there may be security concerns about accessing URL, the workload is quite heavy. However, with more information present about a particular site, it is expected to create a better prediction model.