**Predicting Malicious and Benign Websites**

Hadeel Alrehaili

**Abstract**

One of the biggest challenging on the internet now is web Security; there are billions of websites active on Internet, and threats on the Internet is increasing and the hackers evolving newer techniques to trap web users. With the advancement of technology, especially in the field of AI and ML, we can detect malicious websites by using machine learning.

Therefore, the **Goal** of the project is to build a classification ML model that can detect these attacks based on previous attacks happened.

**Design**

Data was gathered from Web between November 2019 and March 2020 and hosted in public repository. The data were collected from the Internet by scraping webpages using a customized focused web crawler named MalCrawle. The dataset was designed and prepared with the aim of classification of webpages as Malicious or Benign. this dataset contains sufficient information that can be used for any machine learning task related to webpage analysis.

**Data**

The data contains 10000 points with 9 features, 5 of them are categorical (need to be encoding). The features describe each webpage and from these features the page is classify as malicious or benign. IP Address, URL length, geographical location of the IP, protocol used, Domain of the website and many other features that detect whether the website is safe or not.

**Algorithms**

*Feature Engineering*

1. Converting categorical features to numerical variables by encoding
2. Imputing the null value of some features.
3. Handling the missing value and outliers

I build two classification model on a dataset with 10,000 records and 9 features. The entire training dataset was split into 80% for training and 20 for testing. The two model are Logistic regression and random forest classifiers. After training the two models, random forest turns out to be the model with best performance. But in other hand, Logistic regression is doing also good and has a good result in term of accuracy but in overall with other metrics like F1 and recall, random forest is better.

**Final random forest 5 scores (the best overall model):**

* **Accuracy** 0.96
* **F1** 0.91 bad, 0.98 good
* **precision** 0.98 bad, 0.96 good
* **recall** 0.85 bad, 0.99 good

**Tools**

* Jupyter notebook as IDE.
* Pandas and numpy to data processing and cleaning.
* Matplotlib and Seaborn for plotting.
* Scikit-learn for modeling.

**Communication**

Both my presentation and Jupyter Notebook, that contain the complete project will be embedded on my Public repository on GitHub.