PHISHING SITE DETECTION PLUGIN

PHISHTOR

WHAT WE AIM TO DO

- A lite chrome plugin that aims to detect phishing websites and warn the user. It is built with a objective of privacy, so that the user's browsing data need not be collected for classification. The classification is done on the client side with one-time download of the classifier model.
- Real time detection of phishing site.
- Client side classification and thus better privacy.

PROBLEM WE TRY TO SOLVE

- Phishing is defined as mimicking a creditable company's website aiming to take private information of a user. Various phishing website classification techniques use python machine learning libraries and thus they can't be used in the browser in real-time.
- One common approach is to make the prediction in a server and then let the plugin to contact the server for each page. Unlike the old approach, this project aims to run the prediction in the browser itself.

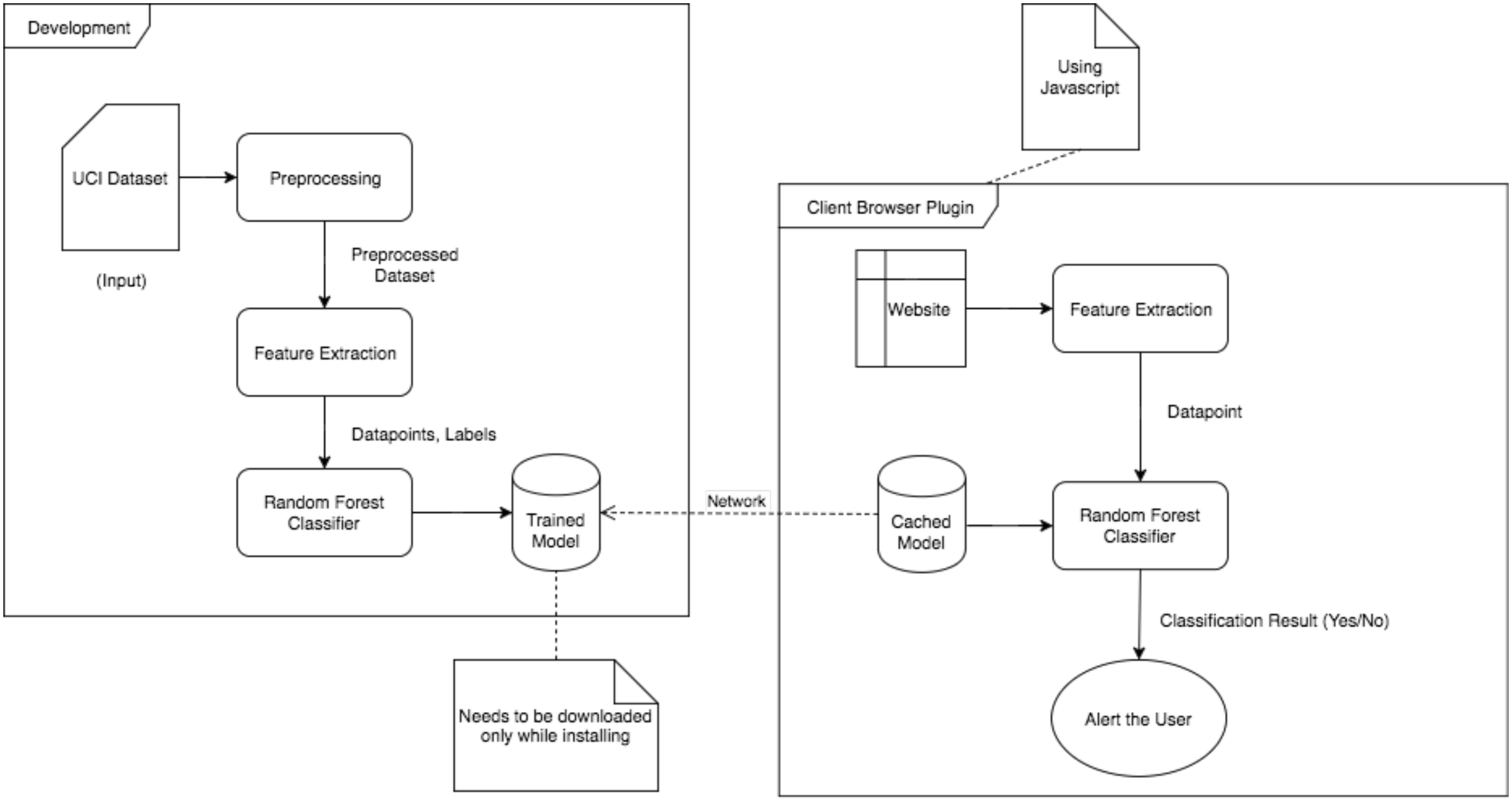
WORKS THAT ARE SIMILAR TO THIS

TITLE	AUTHORS	METHODS USED	RESULTS	FUTURE WORK
Intelligent phishing website detection using random forest classifier (IEEE-2017)	Abdulhamit Subasi, Esraa Molah, Fatin Almkallawi, Touseef J. Chaudhery	Random Forest Classifier	Random Forest has performed the best among the classification methods by achieving the highest accuracy 97.36%.	-
PhishBox: An Approach for Phishing Validation and Detection (IEEE-2017)	Jhen-Hao Li Sheng-De Wang	Ensemble model and a detection model	The false-positive rate of phishing detection is dropped by 43.7% in average.	-
Real time detection of phishing websites (IEEE-2016)	Abdulghani Ali Ahmed Nurul Amirah Abdullah	URLs are inspected based on particular characteristics to check the phishing web pages.	The detection mechanism is capable to detect various types of phishing attacks maintaining a low rate of false alarms.	Automatic detection of the web page and the compatibility of the application with the web browser
Certain investigation on web application security: Phishing detection and phishing target discovery (IEEE-2016)	R. Aravindhan R. Shanmugalakshmi K. Ramya	Various methodologies has been analysed.	An efficient mechanism has been proposed to prevent phishing.	-

HURDLES WE FACE

- **INSTANT PREDICTION** - The user needs to be alerted as quickly as possible before he enters any information in the phishing site.
- **NETWORK LATENCY** - High network latency should not affect the system. Thus depending on a server to make the prediction is going to fail.
- **PRIVACY** - Ensuring user privacy and thus user's browsing data should not leave the user's computer.

HOW IT LOOKS LIKE



PREPROCESSING

- **INPUT:** Dataset in arff format (attribute relation file format)
- **OUTPUT:** Training and testing data in npy format (numpy)
- **METHODOLOGY:** python arff library to load dataset and scikit learn train_test_split for splitting the dataset.
- **EVALUATION METRIC:** cross validation on training set

PSEUDOCODE

```
start
load dataset into numpy array
examine dataset characteristics
split training and test set
export to npy file
end
```

TRAINING

- **INPUT:** Training and testing data npy format (numpy)
- **OUTPUT:** Trained random forest classifier model
- **METHODOLOGY:** Scikit learn Random forest classifier trained on training set and exported in hd5 format.
- **EVALUATION METRIC:** F1 score on testing set

PSEUDOCODE

```
start
load training set from file
calculate cross validation score
train random forest classifier
load testing set from file
Calculate f1 score on testing set
end
```

FEATURE RETRIEVAL – PLUGIN

- **INPUT:** Website viewed by the user.
- **OUTPUT:** 30 features needed for classification.
- **METHODOLOGY:** Javascript DOM API to extract direct features and indirect features are calculated .
- **EVALUATION METRIC:** -

PSEUDOCODE

```
start
obtain URL and DOM of the website
extract url features
extract direct DOM features
calculate indirect features
end
```

PREDICTION – PLUGIN

- **INPUT:** 30 features extracted from the website
- **OUTPUT:** Alert/Popup to the user.
- **METHODOLOGY:** Manual Javascript implementation of the scikit-learn random forest predict function.
- **EVALUATION METRIC:** -

PSEUDOCODE

```
start
load model parameters
load features
reimplement sklearn predict
if seems like phishing
    alert the user
end
```

DATA EVERYWHERE

UCI REPOSITORY: https://archive.ics.uci.edu/ml/datasets/phishing+websites

The dataset is publicly available, labeled and has 11055 records with 30 features.

Data Set Repository:	UCI	**Number of Instances:**	11055	**Area:**	Computer Security
Attribute Characteristics:	Integer	**Number of Attributes:**	30	**Date Donated:**	2015-03-26
Associated Tasks:	Classification	**Missing Values?**	N/A	**Number of Web Hits:**	82911

HOW WE SET TO DEVELOP

- Jupyter Notebook and Python3 with required packages.
- Google chrome plugin development with VS code.
- Github for version tracking https://github.com/picopalette/phishing-detection-plugin

REFERENCES

- Intelligent phishing website detection using random forest classifier - https://ieeexplore.ieee.org/abstract/document/8252051/
- Dataset - https://archive.ics.uci.edu/ml/datasets/phishing+websites