CLOUD BASED PE MALWARE DETECTION API

AI & CYBERSECURITY MIDTERM PROJECT

TABLE OF CONTENTS:

1. Aim :

The main Aim of this Project is to deploy machine learning models for malware classification where we train a neural network and through Amazon Sage Maker deploy the model and classify PE files as malware or benign using Ember opensource dataset, EMBER-2017 v2 available at https://github.com/endgameinc/ember.

2. Requirements :

Here we are using extensively google collab for data processing. After the data processing is done, we are switching on to AWS sage Maker to deploy the model. Also, we will create a normal client by sending a simple normal .exe file to check whether it's malware or not.

3. Implementation :

The Project is implemented step wise or task wise, starting from Training the Model.

3.1. Task-1 (Training):

Training the Model can be done with three steps. They are data extraction & preprocessing, model architecture training and finally Testing the model. It is as follows:

3.1.1. Data Extraction & Preprocessing:

The first and most important step to be taken after downloading your data set is Connecting your google drive to the download collab file.
The data can be downloaded from the below link "https://ember.elastic.co/ember_dataset_2017_2.tar.bz2"
When we are using the collab there is a main drawback which is overloading RAM which causes session crashes. Because of this, it will not be possible to download the data to the runtime every time the session is crashed. Therefore, Google Drive is used to host the data files. Anyways the 15GB provided will not be suffice for this huge data, so here I am installing EMBER and vectorizing the data.

In the Process we may lose the data, so I am saving the files in hdf5 format. Here I am scaling the data not to overload the memory and to increase the accuracy of model.

3.1.2 : Model Architecture and Training:

Here I am designing the architecture of the neural network model in Keras. And next Training the Model. Don’t forget to save the training files into drive in h5 format.
My model is trained with 30 epochs and I got the training accuracy of 97% and Validation accuracy 98%.

3.1.3: Testing the Model:

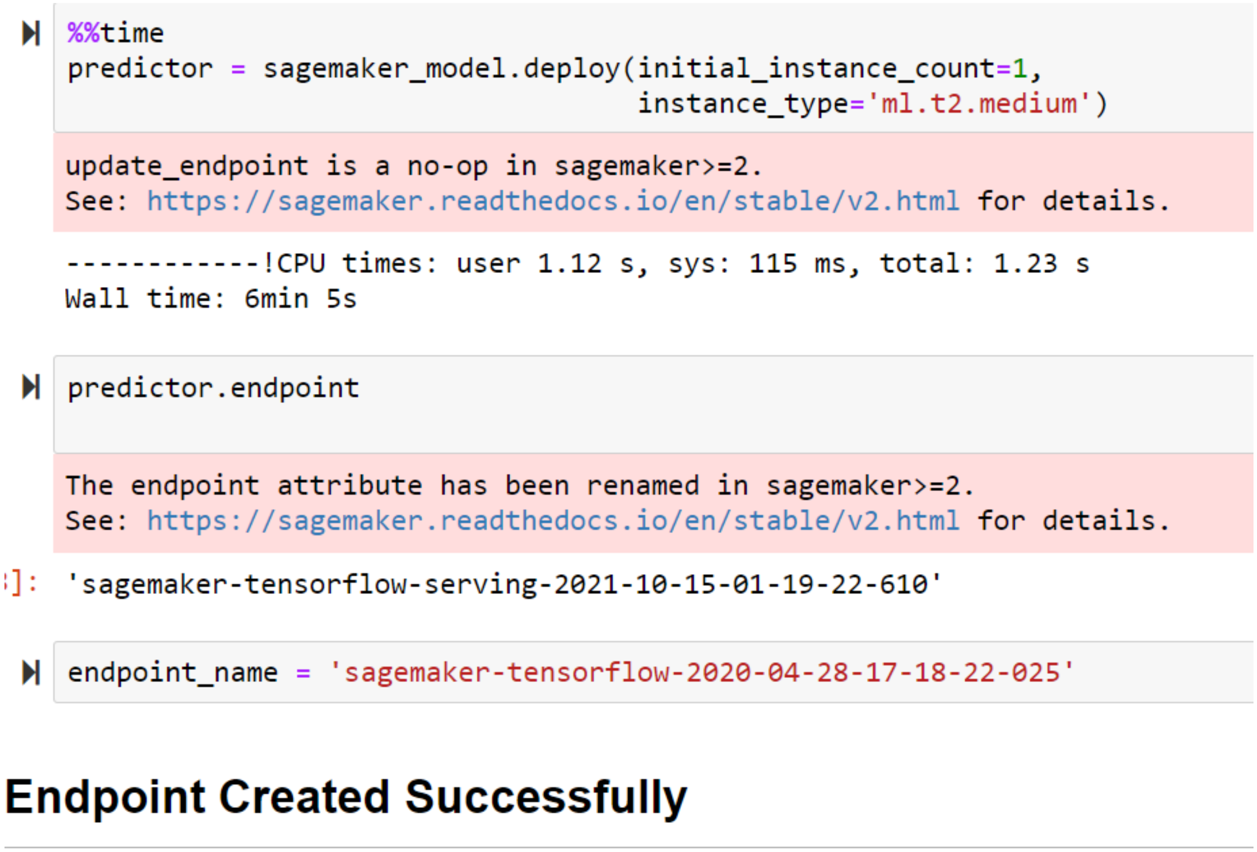
The model is tested on the test data.The trained model is saved and uploaded to Google Drive for future use.

Save the model weights into google drive in h5 format. Also Save your model json into drive, so that we can use this in the next task(deployment). Now test a sample .exe file to check whether it’s benign or malware.

3.2. Task-2 (Deployment of model on the cloud):

After getting access with the AWS Academy and there by management console. Go through the AWS Sage Maker and create a notebook instance with an appropriate conda tensorflow version.

Then, import the model weights, model.h5 and json files into sagemaker and the jupyter notebook. Now deploy the model and create the endpoint and save the name. It will be done in less than 10 minutes.



```
%%time
predictor = sagemaker_model.deploy(initial_instance_count=1,
                                   instance_type='ml.t2.medium')
```

```
update_endpoint is a no-op in sagemaker>=2.
See: https://sagemaker.readthedocs.io/en/stable/v2.html for details.
```

```
------------!CPU times: user 1.12 s, sys: 115 ms, total: 1.23 s
Wall time: 6min 5s
```

```
predictor.endpoint
```

```
The endpoint attribute has been renamed in sagemaker>=2.
See: https://sagemaker.readthedocs.io/en/stable/v2.html for details.
```

```
]: 'sagemaker-tensorflow-serving-2021-10-15-01-19-22-610'
```

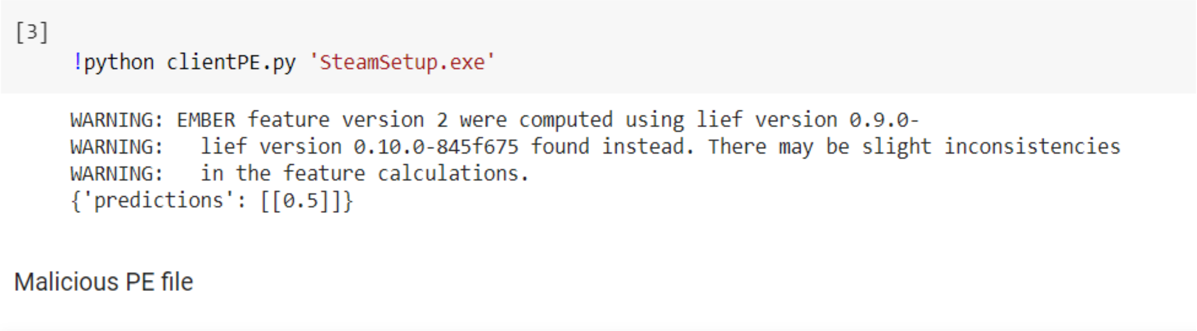
```
endpoint_name = 'sagemaker-tensorflow-2020-04-28-17-18-22-025'
```

Endpoint Created Successfully

3.3. Task-3 (Creating a Client):

Now we are at the end, where we create a client to access the endpoint we created above. For that, we need to get the CLI credentials from AWS and use the endpoint and these CLI credentials to create a client and then test the client.

4. Results: Client Execution Result



```
[3]
    !python clientPE.py 'SteamSetup.exe'
```

```
WARNING: EMBER feature version 2 were computed using lief version 0.9.0-
WARNING:   lief version 0.10.0-845f675 found instead. There may be slight inconsistencies
WARNING:   in the feature calculations.
{'predictions': [[0.5]]}
```

Malicious PE file