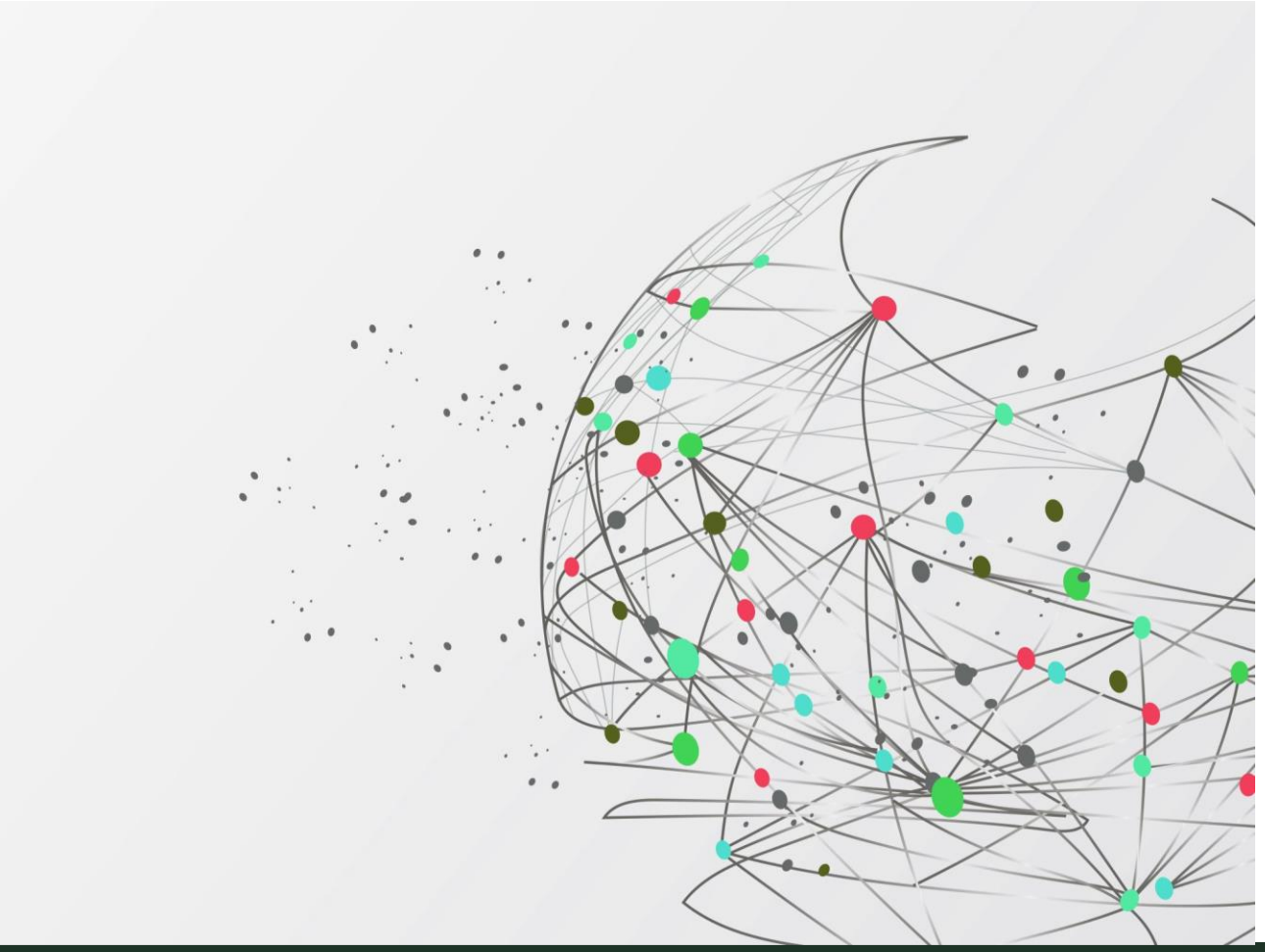


AI & CYBERSECURITY MIDTERM PROJECT

Under Guidance of

Prof. Vahid Behzadan

Mahidar Reddy Uppalapati

CLOUD BASED PE MALWARE DETECTION API

Project's Overview

The purpose of this project is to deploying machine learning models for malware classification. This project comprises of three tasks. The initial task is to train a deep neural network to classify PE files as malware or benign using Ember opensource dataset, EMBER-2017 v2 available at https://github.com/endgameinc/ember.

The second task deals with deploying the model to cloud and creating an endpoint (~API) to the model.

As a final task, create a client nothing but a python script that loads a PE file and classify it as malicious or benign.

The requirements of this project are access to Google CoLab and Amazon Sagemaker. Working in these services is advisable for this project.

Task 1: Training

Mount your Google Drive into your Collab.

Before you start your collab Lab, Please install lief version 0.10.0.

For this project the dataset can be downloaded here. "https://ember.elastic.co/ember_dataset_2017_2.tar.bz2".

Extract all the data and save all data into Google drive.

We need to extract the features from the dataset which are PE files. We deliver this using EMBER LIEF library and vectorize them.

It is known that the EMBER train dataset has three sample categories, namely unlabeled, benign and malicious. As we are interested in classifying Benign and Malicious, We are removing Unlabeled labels.

After data processing, it's common for the collab to crash which requires us to execute everything again from the start. So, it is advised to save the data files to the drive.

Continued..

We save our files in hdf5 format.

Scaling of the data is done using Scikit Learn's different scalars like Standard Scalar, Minmax Scalar and Robust Scalar. Here in this project I am using Standard Scalar.

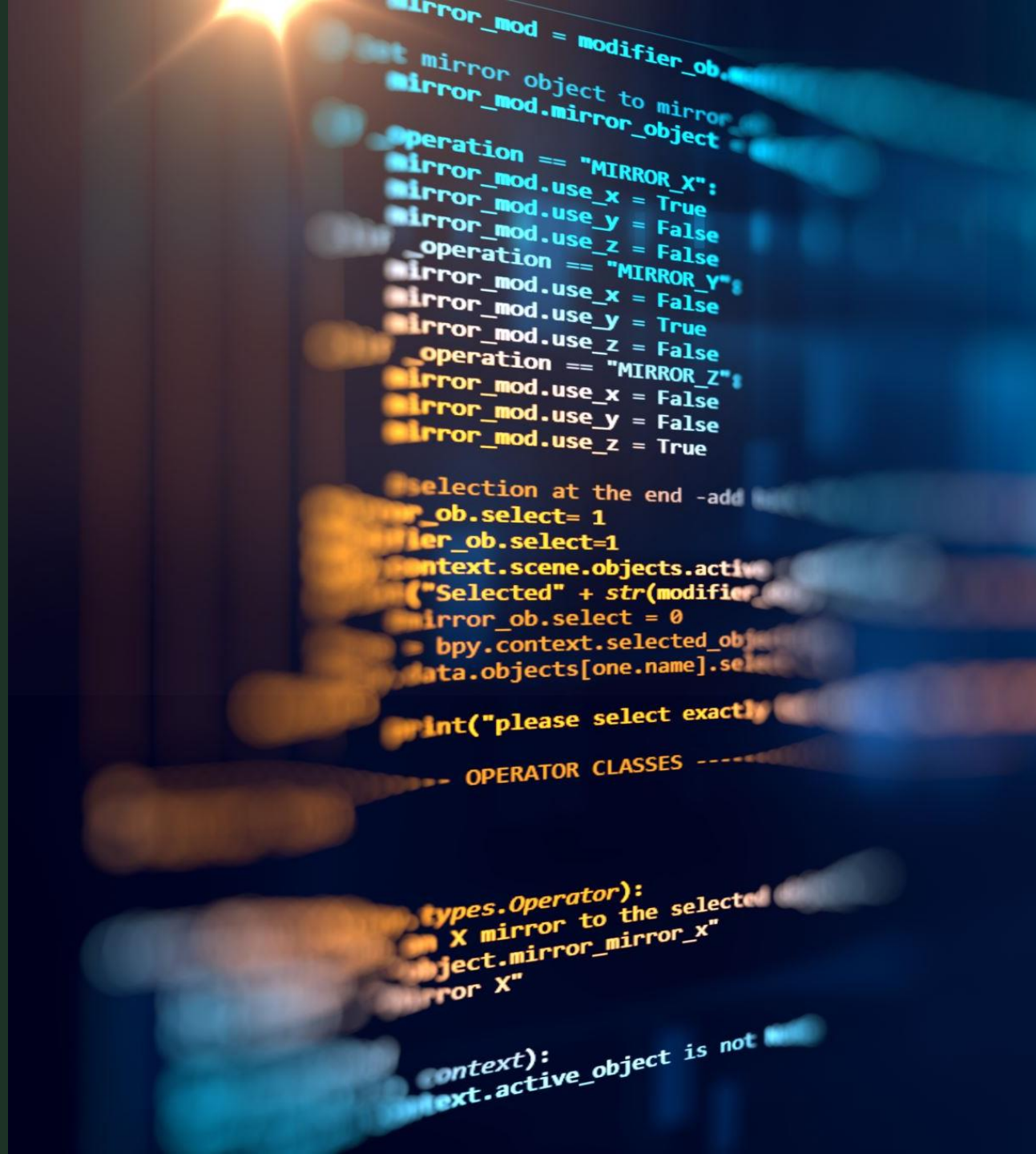
Now we need to design the architecture of the neural network. I build it with a simple and typical dense layers and dropout layers. The dropout layers helps in avoid overfitting the model.

The model is trained with 30 epochs and batch of 256.

Save the model in json format and model weights in h5 format which later be used in deploying the model.

Task 2: Deploying the Model

- To deploy the model to cloud, Create a notebook instance in AWS Sagemaker and create a notebook where all the executions are done.
- Then import the required libraries for the creation of the endpoint for the model.
- Upload the saved model and model weights to the notebook instance. The creation of endpoint took nearly 9 min.
- Endpoint is to be noted as we use it further.



In [20]:
```
%%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 1.16 s, sys: 122 ms, total: 1.29 s
Wall time: 6min 33s
```

In [21]:
```
predictor.endpoint
```

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

Out[21]:
```
'sagemaker-tensorflow-serving-2021-10-15-00-27-55-675'
```

In [ ]:
```
endpoint_name = 'sagemaker-tensorflow-serving-2021-10-15-00-27-55-675'
```

End Point Created successfully

Task 3: Creating a Client

- First check Sagemaker endpoints if it has any problems and check its status.
- Created a client to access the endpoint by using AWS CLI credentials.
- Use the endpoint and proper content type to get your service from the endpoint.
- Testing the Client using a random PE file.

Results

```
!python clientPE.py 'VirusShare_02cbfdd0cf7e38da9a41016dbdb9e291.
```

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

Malicious