DSCI 6015: AI and CyberSecurity

Mid-Term Project

Cloud-based PE Malware Detection API

Ashesh Byanju

Date: 10-13-2021

1. Overview:

The main objective of this project is to train a machine learning model to detect whether the given PE is malware or benign. The project is divided into three tasks. Firstly training a model, deploying it on cloud and generating an api and lastly using that api to classify any given PE file as malware or benign or display the probabilities.

For the dataset I used the EMBER-2017 v2 dataset. Link for the dataset is: (https://github.com/endgameinc/ember).

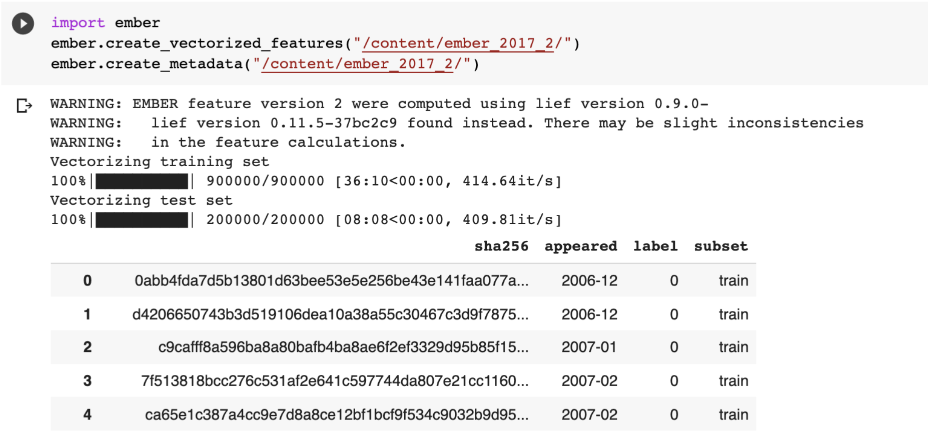
2. Requirements:

For this project we required access to Google Colab and AWS sagemaker since the data is very big and requires good processing power.

3. Approach

3.1 Task 1: Training the model:

This task starts with data loading and preprocessing. I extracted an ember dataset and cloned the github for vectorizing and feature extraction. For this, The LIEF project was used to extract features from the EMBER dataset's PE files. The raw features were converted to JSON and added to the publicly available dataset. From these raw features, vectorized features can be created and saved in binary format, which can then be translated to CSV, dataframe, or any other format.



```
import ember
ember.create_vectorized_features("/content/ember_2017_2/")
ember.create_metadata("/content/ember_2017_2/")
```

```
WARNING: EMBER feature version 2 were computed using lief version 0.9.0-
WARNING:   lief version 0.11.5-37bc2c9 found instead. There may be slight inconsistencies
WARNING:   in the feature calculations.
Vectorizing training set
100%|██████████| 900000/900000 [36:10<00:00, 414.64it/s]
Vectorizing test set
100%|██████████| 200000/200000 [08:08<00:00, 409.81it/s]
```

	sha256	appeared	label	subset
0	0abb4fda7d5b13801d63bee53e5e256be43e141faa077a...	2006-12	0	train
1	d4206650743b3d519106dea10a38a55c30467c3d9f7875...	2006-12	0	train
2	c9cafff8a596ba8a80bafb4ba8ae6f2ef3329d95b85f15...	2007-01	0	train
3	7f513818bcc276c531af2e641c597744da807e21cc1160...	2007-02	0	train
4	ca65e1c387a4cc9e7d8a8ce12bf1bcf9f534c9032b9d95...	2007-02	0	train

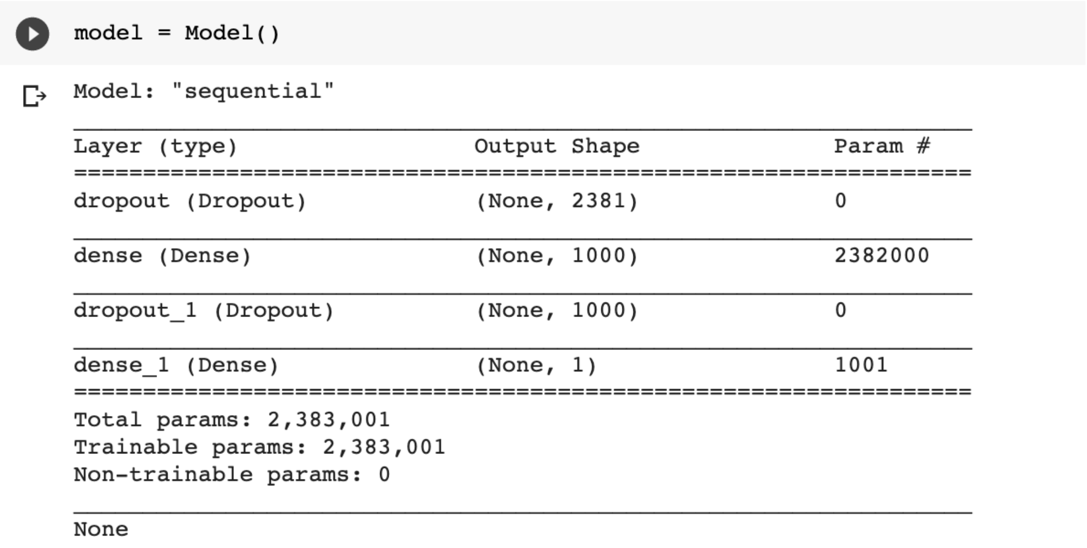
Fig:1 Vectorizing ember dataset

After vectorizing splited the dataset into X-train, y-train, X-test, y-test. Since there were unlabeled data in the dataset I removed such data since they play no role in training and testing.

After removing the data, the training set had 600k data and the test set had 200k data with 2381 features. After this data were normalized. I used Standard scalar at the beginning but the google colab was crashing while running on a training dataset. So, I used a robust scalar and it worked fine for me.

After scaling the dataset, to avoid rerunning the entire process again since the session was crashing a lot of times I created a HDF5 file and saved it in my drive.HDF stands for Hierarchical Data Format and refers to a group of file formats (HDF4, HDF5) that are used to store and organize enormous volumes of data.

To build a neural network using keras I used a simple network with one hidden layer and two drop outs for generalization. Beside this, I used relu for the hidden layer and sigmoid for the output layer. Adam was used as an optimizer, binary cross entropy as loss function and accuracy as performance measure.



```
model = Model()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dropout (Dropout)            (None, 2381)              0
_________________________________________________________________
dense (Dense)                (None, 1000)              2382000
_________________________________________________________________
dropout_1 (Dropout)          (None, 1000)              0
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 1001
=================================================================
Total params: 2,383,001
Trainable params: 2,383,001
Non-trainable params: 0
_________________________________________________________________
None
```

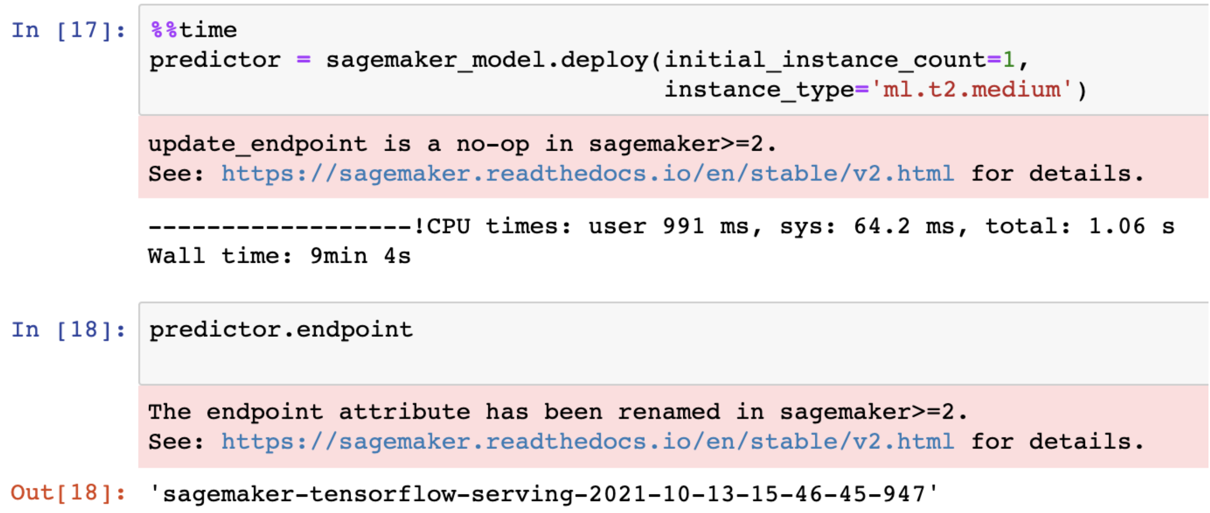
Fig 2: Model Architecture

For training the model, I used 30 epochs with batch size of 256 and splited 20% of the dataset into validation. It took about 2-3 hours to train the model with an accuracy of 83.37%.

For final evaluation I tested it on a test dataset and got an accuracy of 79%.

3.2 Task 2: Deploy the model on the cloud

I created a notebook instance in AWS Sagemaker where all the executions were done to deploy the model to the cloud (AWS). Then, for the creation of the model's endpoint, imported the necessary libraries to the notebook instance, uploaded the stored model and model weights. It took about 9 minutes to create an endpoint.

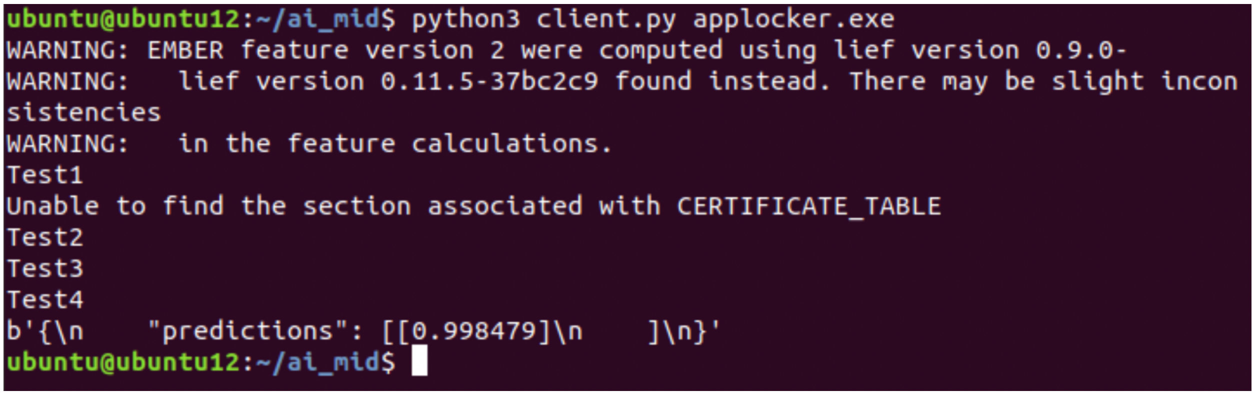


```
In [17]: %%time
         predictor = sagemaker_model.deploy(initial_instance_count=1,
                                            instance_type='ml.t2.medium')

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

         ------------------!CPU times: user 991 ms, sys: 64.2 ms, total: 1.06 s
         Wall time: 9min 4s

In [18]: predictor.endpoint

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

Out[18]: 'sagemaker-tensorflow-serving-2021-10-13-15-46-45-947'
```

Fig 3: Sage Maker Endpoint

3.3 Task 3: Create a client

I created a python file that takes a PE file as an argument, used ember to parse and extract features and vectorize the file, normalize it and use the api created using sagemaker to test whether it is malware or benign. The boto3 library was used to connect to the AWS Sagemaker API, and the required keys and token ids of the AWS CLI were specified.



```
ubuntu@ubuntu12:~/ai_mid$ python3 client.py applocker.exe
WARNING: EMBER feature version 2 were computed using lief version 0.9.0-
WARNING:   lief version 0.11.5-37bc2c9 found instead. There may be slight incon
sistencies
WARNING:   in the feature calculations.
Test1
Unable to find the section associated with CERTIFICATE_TABLE
Test2
Test3
Test4
b'{\n    "predictions": [[0.998479]\n    ]\n}'
ubuntu@ubuntu12:~/ai_mid$ 
```

Fig 4: Client Execution in ubuntu

4. Conclusion

This project has a large dataset and takes high processing power as well as spaces. So, due to free tier google colab and limited access to aws services, it was difficult to execute the project in one go. Despite several crashes and restart of the google colab, I finally managed to get the output. The project was very interesting and I learned a lot.

References

- H. Anderson and P. Roth, "EMBER: An Open Dataset for Training Static PE Malware Machine Learning Models", in ArXiv e-prints. Apr. 2018.
- https://github.com/elastic/ember
- https://youtu.be/8ygCyvRZ074
- https://www.youtube.com/watch?v=8Vj7OaR4DcA
- https://www.youtube.com/watch?v=2_z2kgkt5AM
- https://github.com/aws-samples/amazon-sagemaker-keras-text-classification