Android Malware Detection With Neural Net

•••

By: Ashesh Byanju

Date: Dec 6, 2021

Introduction

- The number and complexity of Android malware has increased, posing serious dangers to the security of mobile devices and the services.
- Data science has become an attractive subject in cybersecurity because analytical models based on data allow for the discovery of insights that might assist forecast dangerous activities
- In this project, I will be training a model using neural network in order to detect android malware.
- Will be utilizing open datasets to evaluate specific network layer features as the foundation for machine learning models that can detect android malware.

Statement Of Project Objectives

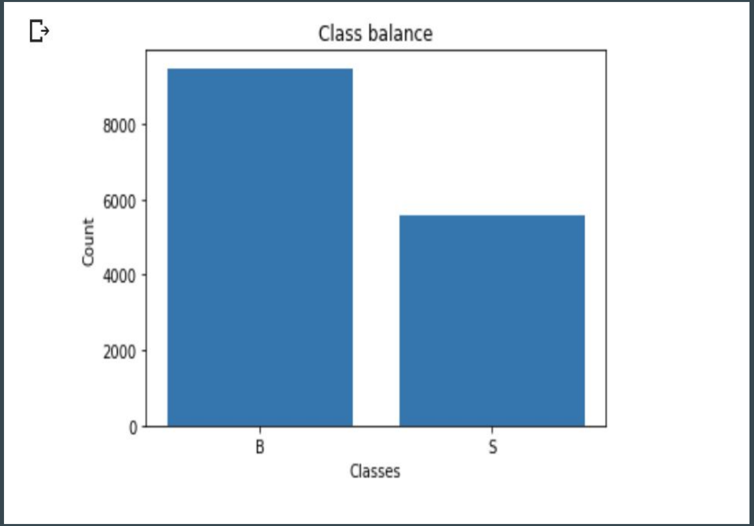
- Dataset is a csv file consisting of information about the android devices that was created from the feature extraction process with DREBIN and Malgenome project malware samples
- Dataset consisting of feature vectors of 215 attributes extracted from 15,036 applications (5,560 malware apps from Drebin project and 9,476 benign apps)
- Link to dataset: https://figshare.com/articles/dataset/Android_malware_dataset_for_machine_learning_2/5854653
- The main goal is to detect if there is presence of malware by using the attributes extracted from Android applications as features
- Will be building neural network using tensorflow to achieve goal
- So, whenever we have new dataset with all the attributes needed, we can clearly detect malware and prevent unauthorized access to privacy sensitive informations.

Approach

- Tools :
  - Google colab
  - Tensorflow
  - Keras
  - Pandas
  - Python 3.7
- Techniques :
  - Detecting malware using artificial neural network using keras
  - After the training the model we will gather our statistics and plot the results to see how model performed and highlight key features.

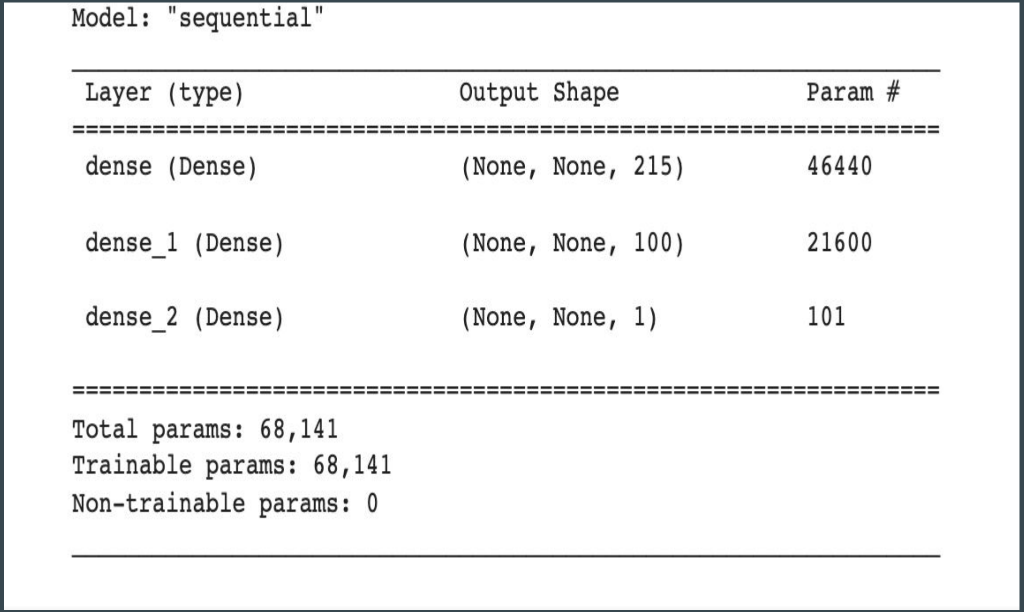
Implementation

- Setting up colab and importing necessary libraries
- Started with data loading and preprocessing
- Remove null values from data
- Converted categorical data into numerical data using Label-Encoder.
- Benign Samples: 9,476
- Spam samples: 5,560



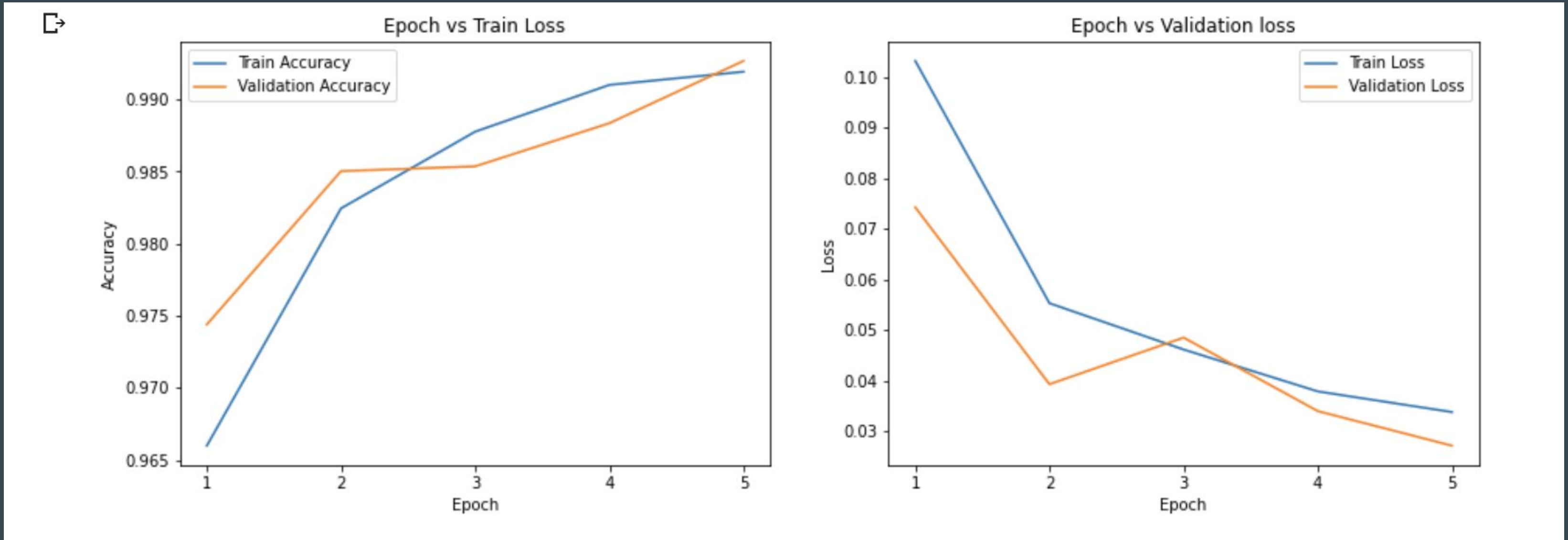
Implementation Continue...

- Model Initialization : neural net
- RMSprop for optimizer
- Binary cross entropy for calculating loss
- Epochs: 5



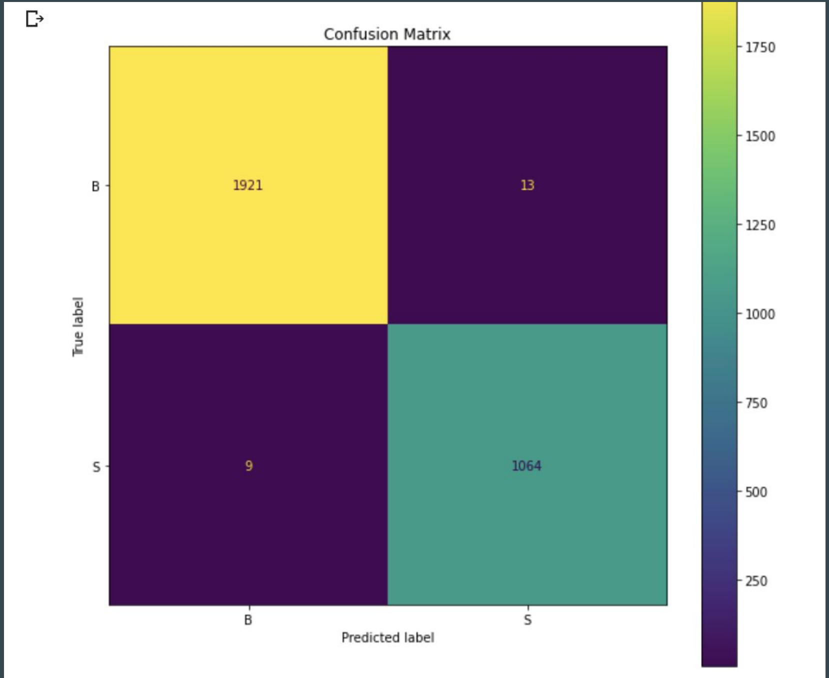
```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, None, 215)         46440

 dense_1 (Dense)             (None, None, 100)         21600

 dense_2 (Dense)             (None, None, 1)           101

=================================================================
Total params: 68,141
Trainable params: 68,141
Non-trainable params: 0
_________________________________________________________________
```

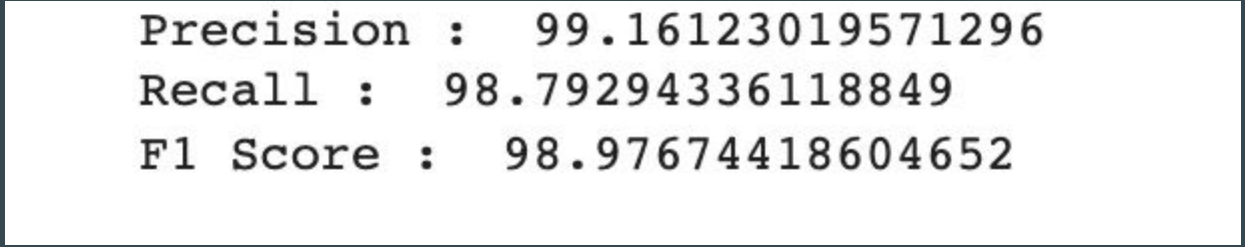
Evaluation And Result



Evaluation And Result

- Accuracy: 99%
- Higher true positive and true negative results





```
Precision :  99.16123019571296
Recall :  98.79294336118849
F1 Score :  98.97674418604652
```

Conclusion

- Accurately classified data as malware or benign
- Accuracy of 99% is a very good result
- Interesting project and a great learning

References

- https://figshare.com/articles/dataset/Android_malware_dataset_for_machine_learning_2/5854653
- https://ieeexplore.ieee.org/document/8245867
-

Thank You!