**AI & CYBER SECURITY MIDTERM PROJECT REPORT**

**(DSCI-6672-01-S20)**

**On**

**CLOUD-BASED PE MALWARE DETECTION API**

**By**

**AKHIL**

**Under the guidance of**

**VAHID BEHZADAN, Ph.D.**



**Tagliatela College of Engineering**

**Department of Data Science**

**300 Boston Post Rd, West Haven, CT 06516**

**ACKNOWLEDGEMENT**

Our most sincere and grateful acknowledgement is due to this sanctum, **University Of Newhaven,** for giving us the opportunity to fulfill our aspirations and for good career.

We express our heartfelt gratitude to our guide **Mr. Vahid Behzadan, Ph.D.** for hisesteemed guidance. We are indebted for his guidelines that proved to be very much helpful in completing our project successfully in time.

Our special thanks to our classmates for providing their special guidance and support for the completion of our project successfully.

**Submitted By**

**Akhil Naraharisetti (00667837)**

**ABSTRACT**

The purpose of this term project is to demonstrate practical skills in implementing and deploying

machine learning models for malware classification. The technical implementation of this project is

comprised of three main tasks that need to be completed sequentially.

**LIST OF CONTENTS**

**DESCRIPTION PAGE NO**

PROJECT TITLE 1

ACKNOWLEDGEMENT 2

ABSTRACT 3

LIST OF CONTENTS 4

PROJECT OVERVIEW5

TECHNICAL APPROACH 6

PERFORMANCE ANALYSIS 10

REFERENCES 12

1. **PROJECT OVERVIEW**

**Task 1 - Training**:

In this task, I will be creating and training a deep neural network based on the MalConv architecture to classify PE files as malware or benign. As for the dataset, you will be using the EMBER-2017 v2 (https://github.com/endgameinc/ember). The model must be implemented in Python 3.x using TensorFlow (1.x or 2.x) and Keras, and needs to be coded and documented in a Jupyter Notebook. : This model may take a long time to train on personal computers (from 7-8 hours to a couple of days, depending on the config), unless you already have a powerful NVIDIA GPU(1080 TI or better).

• **Post-Training:** Once your model is trained, save and store the model. Then, create a function (or method) that takes a PE file as its argument, runs it through the trained model, and returns the output (i.e., Malware or Benign).

**Task 2 - Deploy your model on the cloud:**

In this task, you will be using Amazon Sagemaker to deploy your model on the cloud, and create an endpoint (~ API) so that other applications can make use of the model. While this might sound complicated, you will find that it is actually quite simple to deploy models using Sagemaker.

**Task 3 – Create a client:**

This task is quite simple as well: create a Python code that loads a PE file, converts it into a feature vector that is compatible with your MalConv/EMBER model, runs the vector on the cloud API, and then prints the results (i.e., Malware or Benign – or probabilities of each).

1. **TECHNICAL APPROACH**

I’ve trained the ember dataset which is version 2 of 2017 using train\_ember.py and classify\_binaries.py. It took great amount of time to run where I used my local instance for training as my System’s configurations are good enough for the model to get trained. So, it took around 2 hours for training the data. I have also calculated the process time. In the picture below, you can notice the amount of time it took to train the data. The model used for training was LGB model (Light Gradient Boosting). It is fast in model training and low in memory usage. Also, it is a tree-based learning algorithm. It is easy to use but the only complicated thing is parameter tuning.

A screenshot of a social media post

Description automatically generated

Creating vectorized features and metadata from ember package for splitting the data into train and test. Also, the metadata can be used as a dataframe (pandas) for fitting this into LGB model. The picture below shows creating the vectorized features and the metadata.

A screenshot of a social media post

Description automatically generated

**Using Neural Nets**: I have built the Sequential Keras model using multiple Dense and Activation Layers. Optimizer used is ‘Adam’ and loss is ‘mse’. It took about like 15 minutes to train the model.

**Task 2:** I have 2 models to deploy on Amazon cloud. I’ll be deploying keras model. I have successfully deployed the model on AWS. I have loaded the keras model using the json and h5 files. Later I have exported the model to Tensorflow Protobuf format. This is done in Amazon Sagemaker. Then converted TensorFlow model to a SageMaker readable format. Later, deploy the model.

<https://midtermproject.notebook.us-east-1.sagemaker.aws/notebooks/Mid%20term%20project.ipynb#Load-the-Keras-model-using-the-json-and-weights-file%C2%B6>

A screenshot of a social media post

Description automatically generatedA screenshot of a social media post

Description automatically generatedA screenshot of a social media post

Description automatically generated

1. **PERFORMANCE ANALYSIS**

Calculating the ROC-AUC score and the detection rate at 2 False-positive rates i.e, at 1% and 0.1% FPR. Also calculating their thresholds. This ensures the LGB model performance on Ember dataset. I finally achieved ROC AUC score of 0.999767. The detection rate for malware classes for 1% is 99.56% and for 0.1 % is 98.73%.

A screenshot of a social media post

Description automatically generated

I have used basic evaluation metric which is built-in function of sequential model in keras. I’ve also used Classification accuracy metric imported from sklearn and achieved a very low accuracy of 50.003% after Standardization.

**A screenshot of a social media post

Description automatically generated**

1. **REFERENCES**

* <https://github.com/endgameinc/ember/tree/master/malconv>
* <https://github.com/endgameinc/ember/tree/master/malconv>
* Google search engine