TEMITOPE ADENIYAN

**MIDTERM PROJECT**

**C**loud-Based PE Malware Detection API

**INTRODUCTION**

**Introduction:**

In the realm of cybersecurity, the menace of malware, encompassing various detrimental software and scripts, poses a substantial threat to computer systems, demanding meticulous attention. These malicious entities are adept at executing diverse destructive operations, ranging from data theft to surreptitious surveillance of user activities. Accurate categorization of such malevolent entities is pivotal for devising effective detection and mitigation strategies.

The aim of this project is to showcase proficiency in creating and deploying machine learning models for malware identification. The project comprises three primary objectives: constructing and training the model, and deploying it as a cloud API with Amazon SageMaker.

This report delineates the development and deployment of a neural network employing the MalConv architecture to classify PE files as either malicious or benign. The model was trained utilizing the EMBER-2017 v2 dataset on Colab, and subsequently deployed on Amazon SageMaker with an associated S3 Bucket.

**Task 1: Building and Training the Model:**

Task 1 aimed to develop a deep neural network based on the MalConv architecture for classifying Portable Executable (PE) files as malware or benign. The model was trained on the EMBER-2017 v2 dataset, which comprises over a million PE files annotated with labels indicating their malicious or benign nature.

The MalConv architecture, specifically tailored for malware classification tasks, operates directly on the raw byte sequence of PE files, enabling capture of intricate patterns indicative of malicious behavior. Leveraging convolutional layers and max-pooling operations, MalConv learns hierarchical representations of PE files, facilitating effective classification.

The EMBER-2017 v2 dataset offers a diverse and extensive collection of samples, enabling robust model training and evaluation. Each PE file is represented as a byte sequence, supplemented with metadata such as file size and entropy for analysis.

The technical approach involved several key steps. Firstly, pre-processing of PE files was conducted to extract relevant features and convert them into a format suitable for input into the neural network. Subsequently, the MalConv architecture was implemented using PyTorch, configuring layers and parameters to align with the original design. The model was then trained using supervised learning techniques, employing mini-batch gradient descent and cross-entropy loss. Lastly, various hyperparameters were experimented with to optimize model performance, including learning rate, batch size, and number of epochs.

Training Process:

The training process encountered challenges related to model convergence and overfitting, which were mitigated through meticulous experimentation with hyperparameters and regularization techniques. Early stopping criteria were implemented to prevent overfitting and enhance generalization performance.

**Conclusion:**

This project signifies a notable achievement in implementing and deploying a machine-learning model dedicated to the critical task of malware classification. Despite encountered challenges during the training process, the resultant model demonstrates efficacy in effectively classifying PE files as malware or benign, underscoring the utility of the MalConv architecture for malware detection tasks.

**REFERENCES:**

<https://www.youtube.com/watch?v=MsZmnUO5lkY>

<https://sagemaker-examples.readthedocs.io/en/latest/intro.html>

<https://sagemaker-examples.readthedocs.io/en/latest/frameworks/pytorch/get_started_mnist_train_outputs.html>

<https://docs.aws.amazon.com/sagemaker/latest/dg/deploy-model.html>

<https://github.com/endgameinc/ember/tree/master/malconv>