Title: Malware Classification with AWS SageMaker

AI and Cybersecurity– DSCI-6015-01

Under the guidance of Professor Vahid Behzadan

Manogna Vennela Ramireddy

Student ID: 00877269

MS in Data Science

University New Haven

Date: 03/29/2024

1. Project Purpose:

In today's digital landscape, the proliferation of cybersecurity threats, particularly malware, poses significant risks to individuals, organizations, and society at large. Malware classification is pivotal in identifying and mitigating these threats by accurately categorizing executable files as benign or malicious. Traditional signature-based methods often fall short in detecting new and unknown malware variants, necessitating the adoption of more advanced techniques such as machine learning. This project focuses on developing a machine learning model for malware classification utilizing the capabilities of AWS SageMaker, leveraging cloud computing's scalability and flexibility to create a robust solution capable of handling large-scale datasets and real-time classification tasks.

1.1. Significance of Malware Classification:

Malware poses a significant threat to individuals, organizations, and critical infrastructure worldwide. The ability to accurately classify PE files can aid in proactive threat mitigation, enhancing cybersecurity posture and safeguarding sensitive data from malicious actors. By automating the classification process, security teams can prioritize and respond to potential threats more efficiently, reducing the risk of data breaches and system compromises.

1.2. Scope of the Project:

The project scope encompasses the entire lifecycle of machine learning model development, from data acquisition and preprocessing to model training and deployment. By adopting a comprehensive approach, the project aims to deliver a scalable and reliable solution for malware classification that can adapt to evolving threats in the cybersecurity landscape.

2. Requirements:

To successfully execute this project, access to both Google CoLab and Amazon SageMaker is necessary. These platforms provide the computational resources and services required for training and deploying machine learning models at scale.

2.1. Platform Considerations:

Google CoLab offers a collaborative environment with access to powerful GPU resources, ideal for model training and experimentation. Amazon SageMaker, on the other hand, provides scalable infrastructure for deploying machine learning models in production environments, ensuring high availability and reliability. By leveraging these platforms, the project can achieve optimal performance and scalability while minimizing infrastructure costs and resource overhead.

3. Implementation:

The project implementation is meticulously organized into distinct tasks, each encompassing several subtasks and methodologies.

3.1. Task 1 - Training:

The first task focuses on training a robust machine learning model capable of accurately distinguishing between malicious and benign PE files.

3.1.1.Data Extraction & Preprocessing:

Using a Jupyter notebook hosted on Google CoLab, the EMBER-2017 v2 dataset is extracted and pre-processed. Leveraging the Ember library, features are extracted from PE files and stored in JSON format. The dataset is partitioned into training and testing sets, with careful consideration given to feature scaling to ensure optimal model performance. Additionally, data augmentation techniques may be employed to increase the diversity of the training dataset and improve the model's robustness to unseen malware variants.

3.1.2. Model Architecture & Training:

The architecture of the neural network model is meticulously designed using the Keras framework. Various configurations of dense and dropout layers are explored to strike a balance between model complexity and generalization. The model is trained using different epochs and hyperparameters, with performance metrics such as accuracy closely monitored. Additionally, techniques such as early stopping and learning rate scheduling may be employed to prevent overfitting and improve convergence speed.

3.1.3. Model Testing:

The trained model is rigorously tested on the held-out test dataset to evaluate its performance accurately. A dedicated function is developed to classify PE files based on the learned model, providing insights into its real-world applicability. Model evaluation metrics such as precision, recall, and F1-score are computed to assess the model's effectiveness in correctly identifying malware samples while minimizing false positives.

3.2. Task 2 - Deploy Model on the Cloud:

In the second task, the trained machine learning model is deployed to the cloud infrastructure provided by Amazon SageMaker.

3.2.1. Endpoint Creation:

A notebook instance is instantiated within Amazon SageMaker, facilitating the seamless deployment of the trained model. The model and associated weights are uploaded to the notebook instance, and an endpoint is created to expose the model's capabilities via a scalable API. Considerations such as endpoint configuration, instance type, and endpoint scaling policies are carefully evaluated to ensure optimal performance and cost-effectiveness.

3.3. Task 3 - Create a Client:

The final task involves creating a client application capable of interacting with the deployed model for real-time PE file classification.

3.3.1. Python Script Development:

A Python script is developed to serve as the client application. Leveraging the AWS SageMaker API, the script interacts with the deployed endpoint, enabling users to classify PE files' nature promptly. Error handling mechanisms are implemented to handle edge cases and ensure robustness in real-world scenarios. Additionally, the client application may incorporate features such as batch processing and result visualization to enhance usability and user experience.

4. Conclusion:

Through the systematic execution of the project tasks, valuable insights are gained into the complexities of machine learning model development and deployment for cybersecurity applications. The project not only demonstrates proficiency in model training and deployment but also highlights the practical challenges encountered in real-world scenarios, contributing to a deeper understanding of machine learning integration within the cybersecurity domain. Furthermore, the project lays the foundation for future research and development efforts in the field of malware detection and threat intelligence, paving the way for more effective cybersecurity solutions.