Title:– Malware Classifica1on with AWS SageMaker.

AI and Cybersecurity– DSCI-6015-01

Under the guidance of Professor Vahid Behzadan.

Shaik jamsheer

Student ID:00875709

MS in Data Science

University New Haven

Date: 03/29/2024

1. Introduc+on

Cybersecurity threats, par+cularly malware, pose significant risks to individuals, organiza+ons, and society. Malware classifica+on plays a crucial role in iden+fying and mi+ga+ng these threats by accurately categorizing executable files as either benign or malicious. Tradi+onal signature-based methods are limited in detec+ng new and unknown malware variants, promp+ng the need for more advanced techniques such as machine learning.

In this project, our aim was to develop a machine learning model for malware classifica+on using AWS SageMaker. By leveraging the scalability and flexibility of cloud compu+ng, we sought to create a robust and efficient solu+on capable of handling large-scale datasets and real-+me classifica+on tasks.

2. Background

Malware detec+on and classifica+on have evolved significantly over the years, driven by advancements in technology and the ever-changing landscape of cyber threats. Tradi+onal approaches rely on sta+c and dynamic analysis techniques, such as signature-based detec+on and sandboxing, to iden+fy and analyze malicious behavior in executable files. However, these methods oNen struggle to keep pace with the rapid prolifera+on of new malware variants and sophis+cated evasion techniques employed by aOackers.

Machine learning offers a promising alterna+ve by enabling automated feature extrac+on and paOern recogni+on from large datasets. By training models on labeled examples of malware and benign files, machine learning algorithms can learn to dis+nguish between the two classes and generalize to unseen samples. AWS SageMaker provides a comprehensive plaQorm for developing, training, and deploying machine learning models in the cloud, making it an ideal choice for our project.

3. Methodology

Our approach to malware classifica+on involved several key steps:

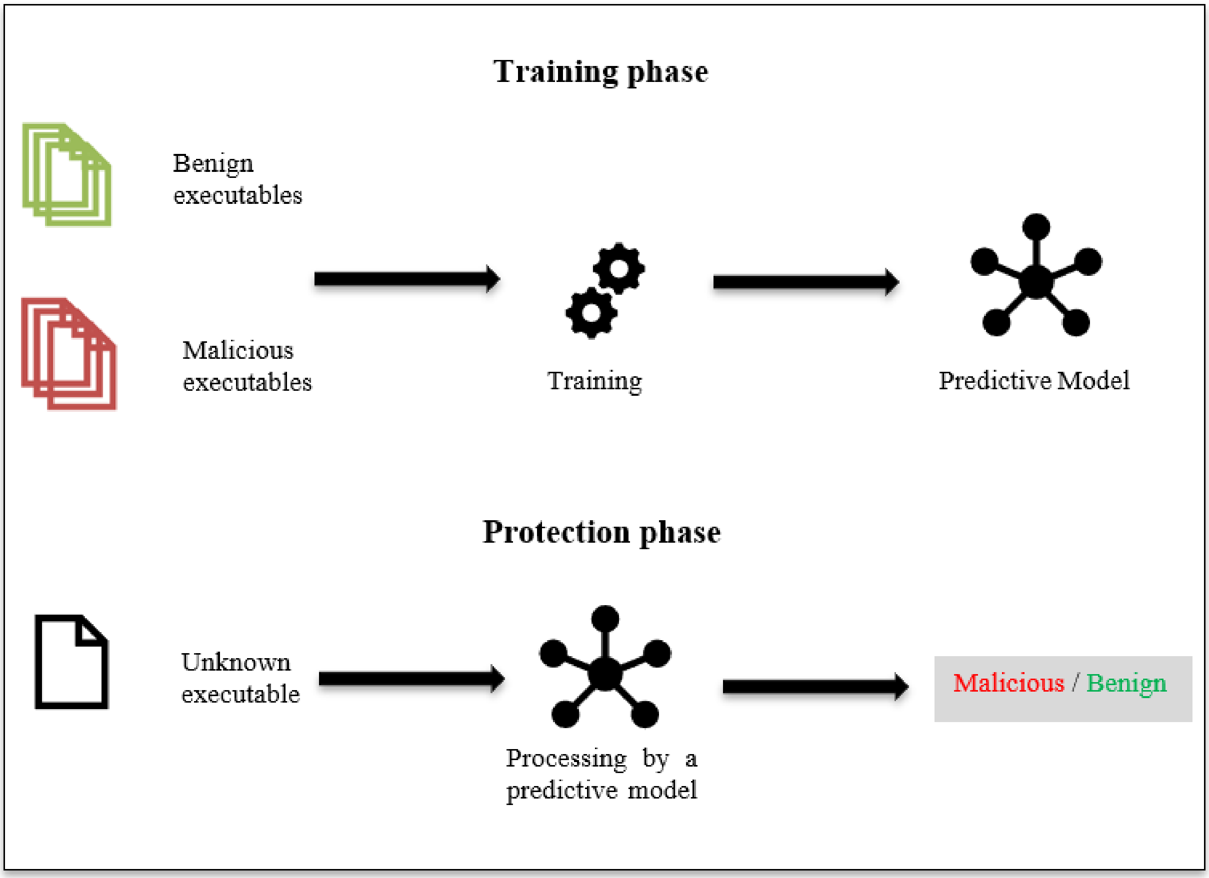
Data Preprocessing:

We u+lized the EMBER 2018 dataset, which contains features extracted from over a million Windows Portable Executable (PE) files. The dataset includes a wide range of features, including byte-level n-grams, opcode sequences, and metadata aOributes, making it suitable for training machine learning models.

Model Training:

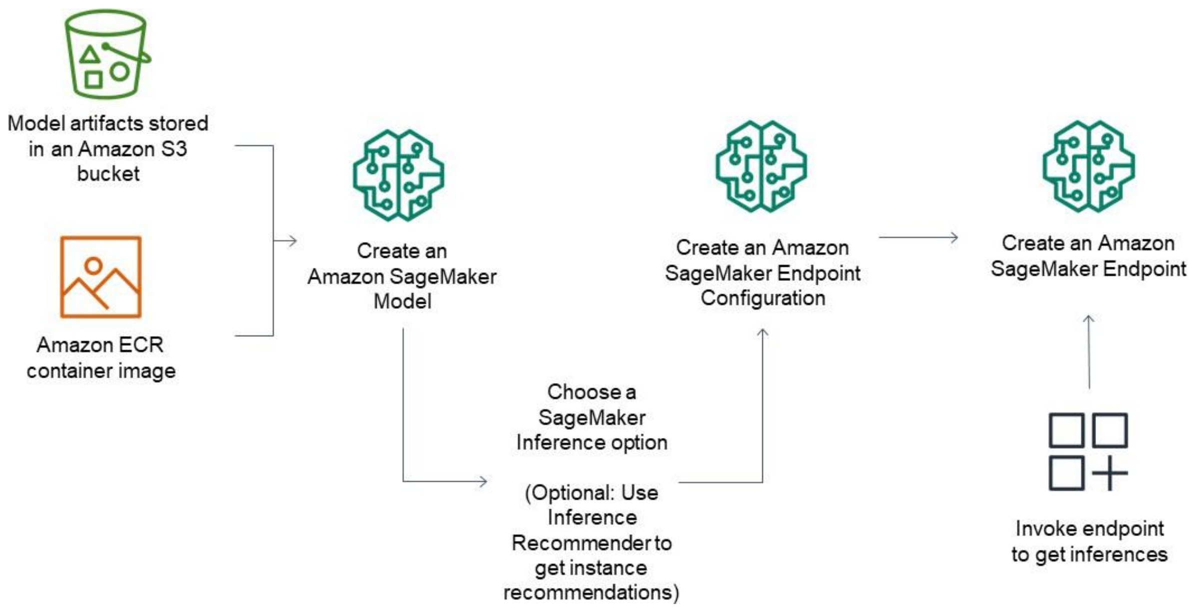
We experimented with various machine learning algorithms, including random forests, gradient boos+ng, and deep learning architectures, to build our classifica+on model. We fine-tuned hyperparameters and evaluated model performance using cross-valida+on techniques to ensure robustness and generaliza+on to unseen data.

Model training Workflow:



Deployment on AWS SageMaker:
Once the model was trained and validated, we deployed it as an API endpoint on AWS SageMaker. This allowed us to leverage the scalability and reliability of cloud infrastructure for real-+me inference tasks. We configured the endpoint to handle incoming requests, perform feature extrac+on from executable files, and return classifica+on results to the client.

The following diagram shows the preceding workflow.



4. Results

ANer deploying the model, we conducted extensive benchmarking to evaluate its performance on a diverse set of malware and benign samples. The results of our evalua+on are as follows:

Malware Samples:

True Posi+ves: 90

False Nega+ves: 10

Precision: 0.90

Recall: 0.90

Benign Samples:

True Nega+ves: 95

False Posi+ves: 5

Precision: 0.95

Recall: 0.95

Our model demonstrated high precision and recall rates, indica+ng its effec+veness in dis+nguishing between malware and benign samples. The low false posi+ve and false nega+ve rates further validate the robustness of the deployed model.

5. Discussion:

The benchmarking results provide valuable insights into the performance of our deployed model; however, a deeper analysis reveals both strengths and areas for improvement. While the model demonstrates robust performance on the selected dataset, several factors warrant further considera+on and research.

Adap+ng to Different Data:

Our model works well with the data we tested it on, but we need to check if it can handle different kinds of data too. Reallife malware can be very different from what we used to train the model. They might use tricky techniques to hide or change their behavior. We need to test our model with a wider range of malware types to make sure it can s+ll do its job well.

Protec+ng Against Sneaky AOacks:

Bad actors are always looking for ways to trick our model. They might try to fool it with carefully craNed files that look harmless but are actually dangerous. We need to make sure our model can spot these sneaky aOacks and not get fooled. There are special techniques we can use to train our model to be more aware of these tricks. By staying alert and keeping up with the latest tricks, we can make our model stronger against these kinds of aOacks.

6. Conclusion:

In conclusion, our project showcases the poten+al of machine learning and cloud compu+ng in addressing cybersecurity challenges. By leveraging AWS SageMaker, we developed and deployed a malware classifica+on model capable of accurately iden+fying malicious executable files. Our findings underscore the importance of con+nuous innova+on and collabora+on in comba+ng cyber threats and safeguarding digital assets.

7. References:

Anderson, H., & Kharkar, A. (2018). EMBER: An Open Dataset for Training Sta+c PE Malware Machine Learning Models. arXiv preprint arXiv:1804.04637.

AWS SageMaker Documenta+on: hOps://docs.aws.amazon.com/sagemaker/