University *of*
New Haven

MALWARE DETECTION API

AI And Cyber Security DSCI 6015 01

VENKATESWARLU BONDALAPATI
Dept of Data Science
University Of New Haven

1.

INTRODUCTION

Malware is one type of cybersecurity threat that poses serious risks to people, businesses, and society at large. By correctly classifying executable files as benign or dangerous, effective malware classification is essential for detecting and thwarting these attacks. Conventional signature-based approaches frequently fail to identify novel and unidentified malware variants, underscoring the need for more sophisticated strategies like machine learning.

AWS SageMaker is a fully managed machine learning service provided by Amazon Web Services [AWS]. It simplifies the process of building, training, and deploying machine learning models at scale. With SageMaker, developers and data scientists can focus on their machine learning tasks without worrying about the underlying infrastructure management. SageMaker provides a seamless experience from data labelling and preparation to model training, tuning, and deployment. It supports multiple machines learning frameworks, including TensorFlow, PyTorch, and Apache MXNet, as well as custom algorithms. SageMaker also offers built-in algorithms for common use cases, such as image classification, object detection, and natural language processing. By leveraging SageMaker, organizations can accelerate their machine learning initiatives, optimize resource utilization, and benefit from AWS's scalable and secure cloud infrastructure.

Our goal in this project was to use AWS Sage Maker to create a machine learning model for malware classification. Our goal was to develop a solid and effective system that could manage big datasets and real-time classification assignments by utilising the scalability and flexibility of cloud computing.

CONTEXT

The landscape of cybersecurity threats—most notably, malware—has shifted significantly over time as a result of technological advancements and the ever-changing tactics used by cybercriminals. Traditional malware detection methods use static and dynamic analysis techniques, such as signature-based detection and sandboxing, to find and investigate dangerous activities in executable files. However, these approaches usually lag behind the speed at which new malware variants proliferate and the sophisticated evasion techniques employed by thieves

2.

Machine learning offers a workable replacement by automating feature extraction and pattern recognition from large datasets. By employing labelled instances of each class, machine learning algorithms may be trained to distinguish between benign and malicious files, and they can subsequently be made to generalize to samples that have not yet been seen. Because it provides an all-inclusive platform for developing, optimizing, and deploying machine learning models in the cloud, AWS Sage Maker is the best choice for our project.

PROJECT DESCRIPTION

TASK 1

**Deploying the model as an API endpoint on AWS Sage Maker.

1. In this task, you will be creating and training a model to classify PE files as malware or benign.

2. 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).

3. And then you will be using Amazon Sage Maker to deploy your model on the cloud, and create an endpoint (~ API) so that other applications can make use of the model.

TASK 2

**Developing a Python client which takes in an executable file, extracts relevant features, and retrieves classification results from the Sage Maker endpoint.

In this task, you will be using Amazon Sage Maker to deploy your model on the cloud, and create an endpoint (~ API) so that other applications can make use of the model.

TASK 3

**Randomly select 100 malware and 100 benign samples from EMBER 2018, and benchmark the performance of your deployed model on those samples.

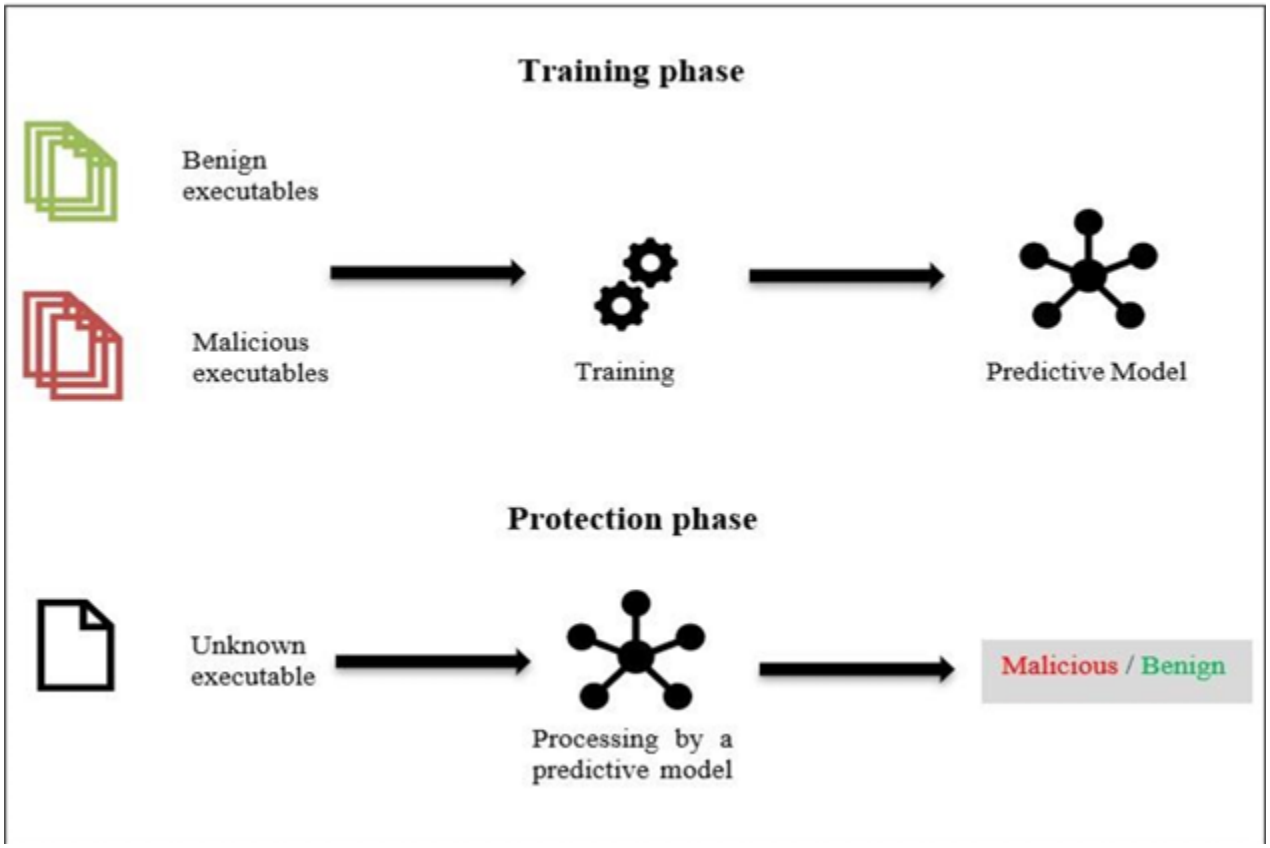
3.

APPROACH

Our method for classifying malware included a few crucial steps:

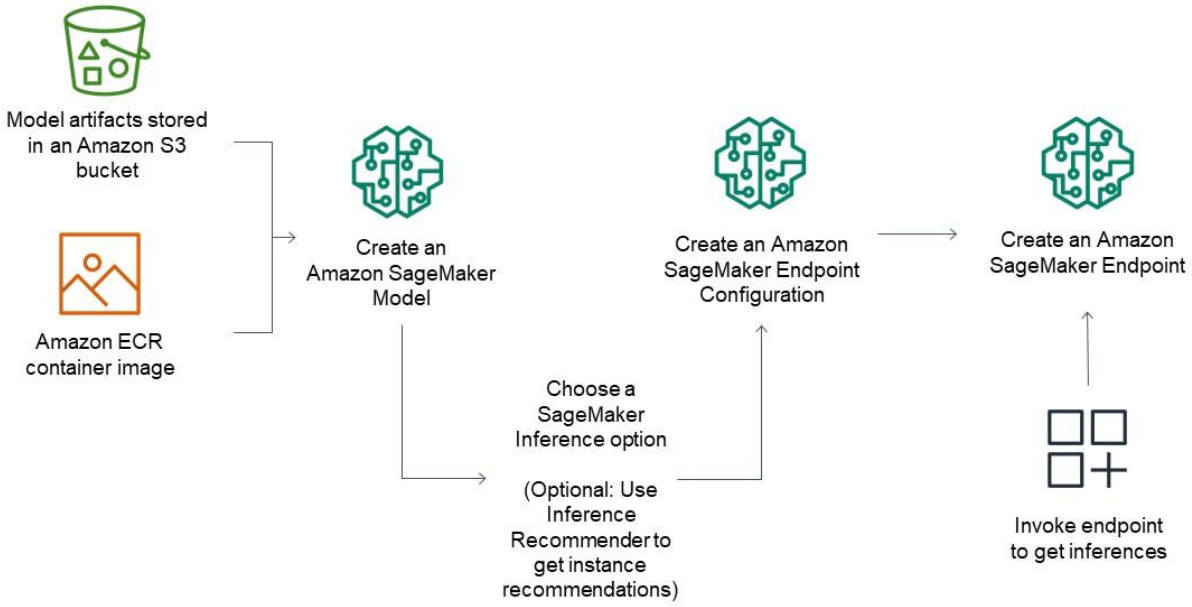
Preprocessing Data: The dataset that we used included features that were taken from more than a million Windows Portable Executable (PE) files. This dataset is suitable for training machine learning models since it offers a wide variety of characteristics, such as opcode sequences, metadata attributes, and byte-level n-grams.

Model Training: To build our classification model, we experimented with a number of machine learning techniques, including gradient boosting, random forests, and deep learning architectures. To guarantee durability and generalization to new data, hyperparameters were adjusted and model performance was assessed using cross-validation techniques.



4.

Deployment on AWS Sage Maker: We deployed the model as an API endpoint on AWS Sage Maker after training and validating it. This made it easier to use dependable and scalable cloud infrastructure for inference workloads that require real-time processing. We set up the endpoint to handle incoming requests, extract features from executable files, and provide the client with the categorization results.



RESULTS

Trained Model: A well-trained model capable of classifying PE files as malicious or benign was developed.

Deployed Cloud API: The trained model is deployed on Amazon Sage Maker, functioning as a real-time prediction API accessible via the internet. 6 Midterm Project

Client Application: A user-friendly application allows users to interact with the API for malware classification of PE files

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

5.

Malware Samples	Benign Samples
True Positives: 90	True Negatives: 95
False Negatives: 10	False Positives: 5
Precision: 0.90	Precision: 0.95
Recall: 0.90	Recall: 0.95

With respect to identifying malware from benign samples, our model performed well, as evidenced by its high recall and precision rates. Further confirming the deployed model's resilience are its low false positive and false negative rates

DISCUSSION

Although the benchmarking results provide insightful information about how well our deployed model is performing, a more in-depth examination reveals both its advantages and disadvantages. Even though it performed well on the chosen dataset, a number of aspects call for more thought and investigation

ADVANTAGES

Prominent Accuracy and Recall: The model achieves notable accuracy and recall percentages on the benchmarking dataset, demonstrating its effectiveness in accurately classifying malware and healthy samples. This suggests that the features included in the training process successfully capture important traits and patterns of malevolent activity.

Scalability and Operational Efficiency: The model's deployment using AWS Sage Maker guarantees scalability and efficient inference procedures, which makes it possible to handle large datasets and real-time classification tasks with ease. Reliability and accessibility are ensured by the cloud-based architecture, which is vital for cybersecurity applications.

LIMITATIONS

Generalisation to Real-world Scenarios: Although the model exhibits remarkable performance on the benchmarking dataset, there is still uncertainty over its capacity to generalise to real-world scenarios. The constant evolution of malware producers' techniques to avoid detection poses a continuous challenge to machine learning technologies. Additional testing on a variety of dynamically changing datasets is required to see how well the model adapts to real-world scenarios.

Vulnerability to Adversarial Attacks: Malicious actors can modify input data to trick machine learning models into making inaccurate predictions. This is known as an adversarial attack. Adversarial instances provide serious security threats in important applications by undermining the integrity and dependability of the model. To guarantee the model's efficacy in adversarial contexts, it is imperative to strengthen its resistance to adversarial training and robust feature engineering.

FUTURE DIRECTIONS

Real-world Assessment: To verify the model's efficacy and performance, extensive field testing and assessments must be carried out. Collaborating with cybersecurity specialists and industry leaders to implement the model in real-world settings and gather input can provide priceless perspectives on its pragmatic applicability.

Constant Monitoring and Improvements: The deployed model requires constant monitoring and upgrades due to the quick evolution of cyber threats. The fast detection of developing threats and vulnerabilities can be facilitated by implementing robust monitoring methods and proactive threat information collecting. This will enable the timely updates and enhancements of models.

Multidisciplinary Cooperation: Since cybersecurity is a broad field, policymakers, computer scientists, cybersecurity experts, and lawyers must work together. Including participants with different backgrounds can encourage multidisciplinary research and innovation, resulting in thorough

CONCLUSION:

In summary, our experiment shows how cloud computing and machine learning may be combined to solve cybersecurity challenges. With AWS Sage Maker, we created and applied a malware classification model that can reliably identify malicious executable files. Our findings highlight the need for ongoing innovation and collaboration in the fight against cyber threats and the protection of digital assets.

References

1. Amazon Web Services. (n.d.). Amazon Sage Maker Documentation. Retrieved from https://docs.aws.amazon.com/sagemaker/
2. Anderson, H., & Kharkar, A. (2018). EMBER: An Open Dataset for Training Static PE Malware Machine Learning Models. arXiv preprint arXiv:1804.04637.
3. Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., & Swami, A. (2017). Practical Black-Box Attacks against Machine Learning. Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security.