University of New Haven

AI and CyberSecurity

DSCI6015

Simplified Midterm Project

Jagadeesh Chandra Bose Mende

00869339

Dr. Vahid Behzadan

April 02 2024.

SUMMARY

This report outlines the successful development of a cloud-based Portable Executable (PE) static malware detection API using AWS Sagemaker. The API employs a Random Forest binary classifier trained on a labeled dataset of binary feature vectors to classify PE files as either malicious or benign.

The project utilized AWS Sagemaker for both model construction and training, as well as for deploying the model. Additionally, a user-friendly web application was developed to allow remote users to upload their executable (.exe) files and assess potential threats. Python was the primary language used for the project, with machine learning libraries such as sklearn, pefile, and nltk employed for model creation and implementation.

INTRODUCTION

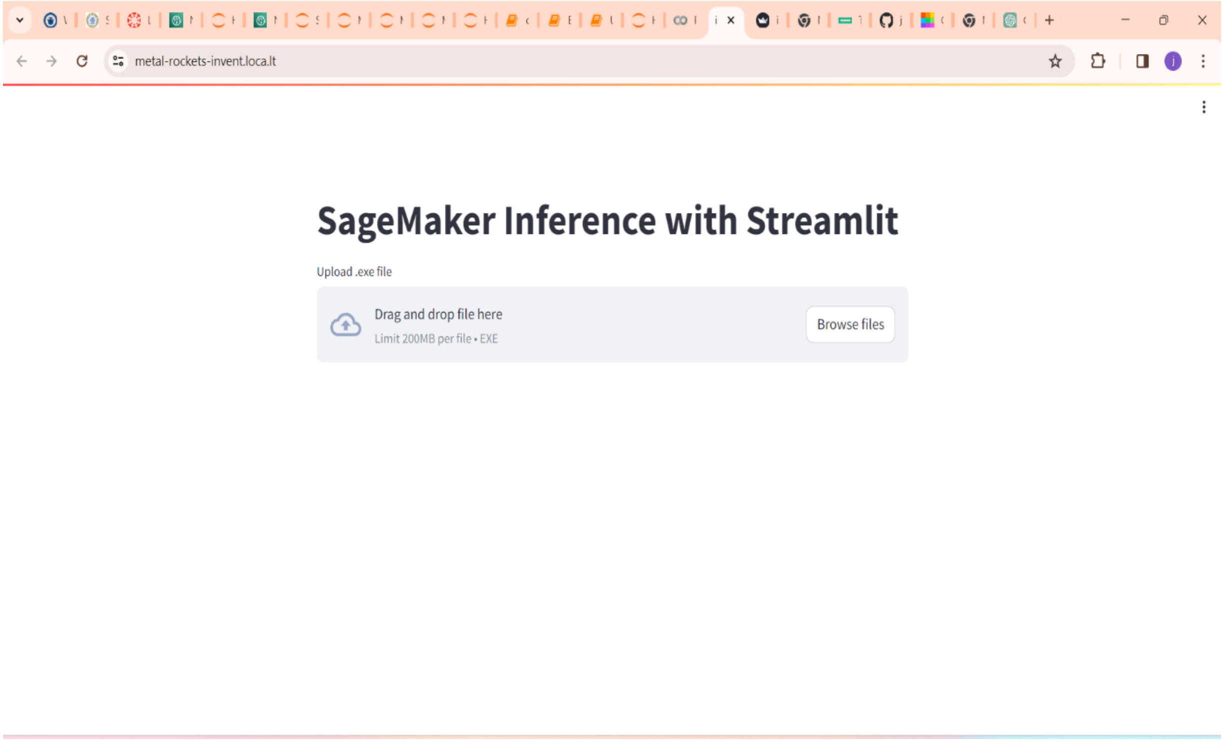
Executable files, known as PE files, are crucial components of Windows operating systems, serving as a standardized format for storing executable code and associated data. These files contain essential information required for program execution, encompassing machine instructions, resources, imported libraries, and metadata. Widely utilized for applications, drivers, and dynamic link libraries (DLLs), PE files adhere to a structured layout characterized by headers that provide insights into the file's attributes, including its architecture, entry point, and section arrangement. Proficiency in understanding the PE file format is invaluable for various tasks such as software analysis, reverse engineering, and malware detection, facilitating the examination and modification of executable content.

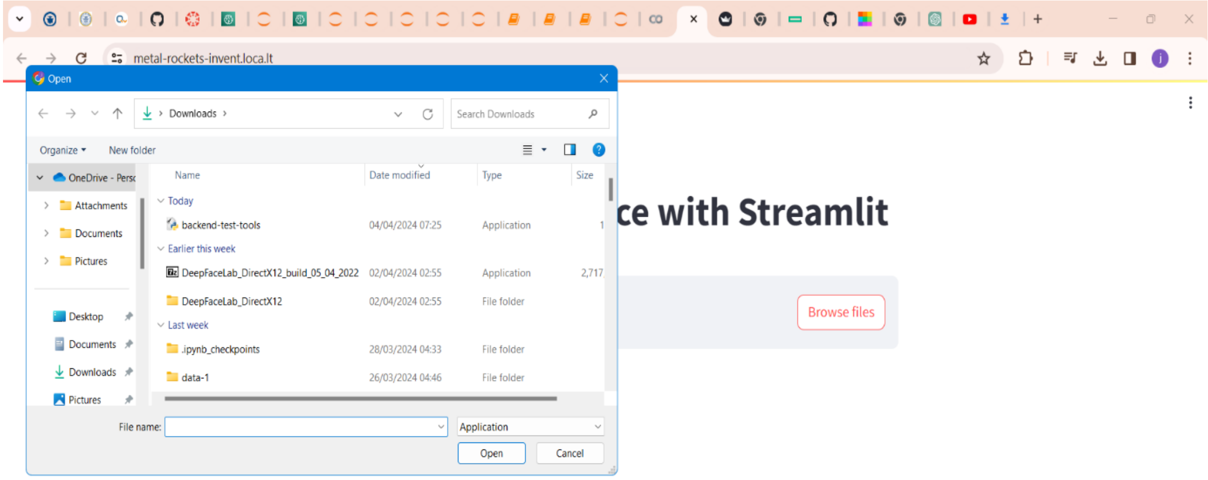
APPROACH SUMMARY:

Model Building and Training: Utilizing an instance of scikit-learn version 1.2.1 on AWS Sagemaker, we trained a Random Forest binary classifier using a meticulously labeled dataset of binary feature vectors. Subsequently, the trained model was saved into a joblib file.

Model Deployment as a Cloud API: Leveraging Amazon Sagemaker, we deployed the trained model to create an endpoint, thereby establishing a cloud-based API for real-time predictions. The model was loaded from the saved joblib file and deployed using the sagemaker.SKLearn module. This involved configuring and creating an endpoint to host the model.

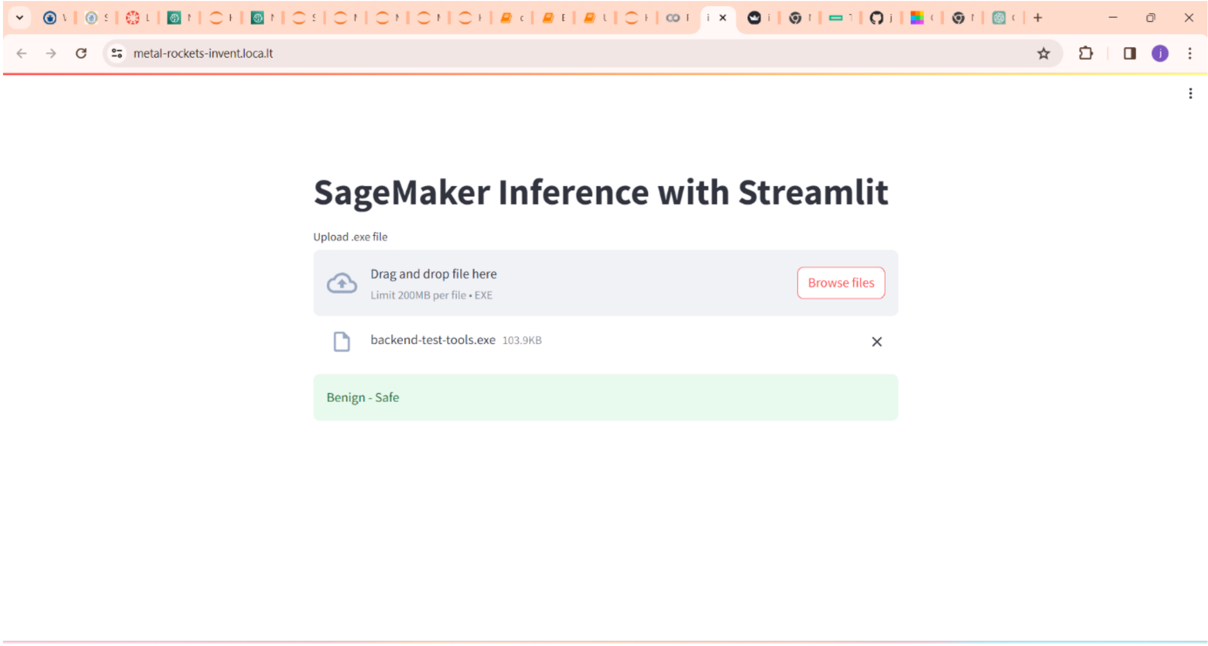
Client Application Development: A Streamlit web application was developed to furnish users with an intuitive interface. Through this application, users can upload executable (.exe) files. The application, utilizing pefile library and other pre-trained data, extracts requisite features from the uploaded .exe files, converts them into JSON format, and dispatches them to the deployed API. The application then showcases the classification results, discerning between Malware (Danger) and Benign (Safe) files. For client deployment, Google Colab was utilized to initiate the Streamlit application.


metal-rockets-invent.loca.lt
SageMaker Inference with Streamlit
Upload .exe file
Drag and drop file here
Limit 200MB per file • EXE
Browse files



Name	Date modified	Type	Size
Today			
backend-test-tools	04/04/2024 07:25	Application	1
Earlier this week			
DeepFaceLab_DirectX12_build_05_04_2022	02/04/2024 02:55	Application	2,717
DeepFaceLab_DirectX12	02/04/2024 02:55	File folder	
Last week			
.ipynb_checkpoints	28/03/2024 04:33	File folder	
data-1	26/03/2024 04:46	File folder	

File name:

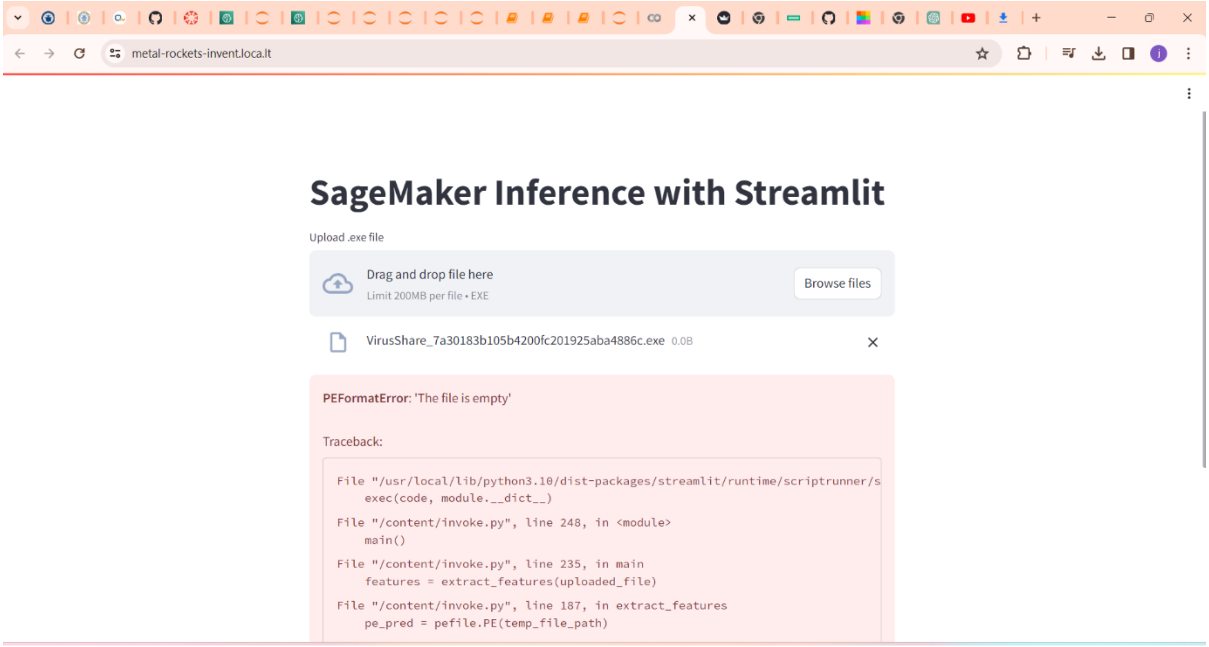


SageMaker Inference with Streamlit

Upload .exe file

backend-test-tools.exe 103.9KB

Benign - Safe



SageMaker Inference with Streamlit

Upload .exe file

Drag and drop file here

Limit 200MB per file • EXE

VirusShare_7a30183b105b4200fc201925aba4886c.exe 0.0B

PEFormatError: 'The file is empty'

Traceback:

```
File "/usr/local/lib/python3.10/dist-packages/streamlit/runtime/scriptrunner/s
    exec(code, module.__dict__)
File "/content/invoke.py", line 248, in <module>
    main()
File "/content/invoke.py", line 235, in main
    features = extract_features(uploaded_file)
File "/content/invoke.py", line 187, in extract_features
    pe_pred = pefile.PE(temp_file_path)
```

RESULTS

The project has effectively delivered its desired results:

Developed a proficient Malware detection model that can classify PE files as either malicious or benign after thorough training.

Deployed the trained model on Amazon Sagemaker, establishing a real-time prediction API accessible via the internet.

Designed a web-based interface for end-users to upload files and verify their malicious status.

CONCLUSION

The obtained accuracy of 99.58% (0.995827770439428) suggests that the model accurately predicts the type of exe file most of the time, achieving a high level of correctness in its predictions.