Title: S24 – Simplified Midterm

AI and Cybersecurity

DSCI:6015-01

Agasti Uday Shankar :: ID - 00890175
Data Science
University of NewHaven
Date: 04/03/2024

Abstract:

This report details the deployment of a machine learning model on AWS SageMaker to classify executable files as benign or malicious. It emphasizes the importance of preserving model integrity and efficient deployment methods. Through the use of joblib, we preserve the model state, enabling seamless integration into real-world applications. Key findings include successful feature extraction from executable files and the development of a user-friendly client application for malware detection. This study highlights the significance of cloud-based deployments in enabling scalable and accessible machine learning solutions for practical applications.

Introduction:

we focus on deploying a machine learning model on AWS SageMaker to classify executable files as benign or malicious. The experiment's significance lies in its practical application, addressing the growing need for efficient malware detection methods in today's digital landscape. With traditional detection methods falling short in combating evolving threats, machine learning offers a promising avenue for enhancing cybersecurity measures. Through this experiment, we aim to showcase the feasibility and effectiveness of deploying machine learning models for real-world cybersecurity applications, highlighting the potential for scalable and accessible solutions in the field.

Methods:

To deploy the machine learning model for classifying executable files as benign or malicious, we employed the Random Forest algorithm. Below are the steps outlining the methodology:

Data Preprocessing:

We begin by preprocessing the dataset containing executable file samples. This involves extracting relevant features from the files, such as byte sequences and metadata.

Feature Engineering:

Next, we perform feature engineering to transform the raw data into a format suitable for training the Random Forest model. This may include encoding categorical features, scaling numerical features, and selecting relevant features using techniques like information gain or correlation analysis.

Model Training:

With the preprocessed dataset, we proceed to train the Random Forest model. The algorithm constructs multiple decision trees based on random subsets of the dataset, using bootstrapping and feature randomness to promote diversity among the trees.

Model Deployment:

After satisfactory evaluation, we deploy the trained Random Forest model on AWS SageMaker. This involves serializing the model and associated artifacts into a deployable format, such as a joblib file, and configuring the SageMaker endpoint to serve predictions.

By following these steps, one can replicate the deployment of the Random Forest model for classifying executable files on AWS SageMaker. This methodology ensures a systematic approach to model deployment while providing flexibility for customization and optimization as needed.

Client Application Development:

Purpose

The client application serves as the interface through which users interact with the system. It provides a user-friendly environment for uploading executable files and receiving analysis results. The key functionalities of the client application include:

File Upload: Users can upload executable files (.exe) to the system for analysis.

User Interface: The client application offers an intuitive user interface that guides users through the analysis process.

Communication with Server: It communicates with the server to send uploaded files and receive analysis results.

Feedback Mechanism: Users are provided with feedback on the analysis process, such as progress indicators and notifications.

Technologies Used

The client application is developed using the following technologies:

Streamlit: A Python library for creating interactive web applications with minimal effort.

Boto3: The AWS SDK for Python, used for interacting with AWS services such as SageMaker.

PEfile: A Python module for working with Portable Executable (PE) files, which are commonly used in Windows-based systems.

Implementation Details

The client application is implemented using the Streamlit framework, which allows for rapid development of web-based user interfaces. It utilizes the Boto3 library to communicate with the SageMaker service for malware analysis. PEfile is employed for extracting features from uploaded executable files.

Upon launching the client application, users are presented with an interface where they can upload files. Once a file is uploaded, the application initiates the analysis process by sending the file to the server for feature extraction and classification. Users receive feedback on the analysis progress and are notified when the analysis is complete.

The client application plays a crucial role in facilitating user interaction with the system, offering a seamless experience for uploading files and receiving analysis results.

Results:

App Interface:

Checking with Benign Sample:

Checking with Malware Sample:

The project successfully achieved its intended outcomes:

Trained Malware Detection Model

A robust malware detection model was developed through extensive training on a diverse dataset of PE files. Leveraging machine learning techniques, the model exhibits high accuracy in classifying PE files as either malicious or benign based on their features.

Deployed Cloud API

The trained malware detection model is deployed on Amazon SageMaker, providing a scalable and reliable infrastructure for hosting machine learning models. The model functions as a real-time prediction API accessible over the internet, enabling seamless integration into various applications and systems.

Web Client

A user-friendly web-based interface was created to facilitate easy interaction with the malware detection system. End-users can securely upload executable files through the web client and receive immediate feedback on the files' maliciousness. The interface offers a streamlined experience for users to check the security status of their files without the need for technical expertise.

Impact

The developed solution contributes to enhancing cybersecurity measures by empowering users to proactively identify and mitigate potential malware threats. By combining advanced machine learning algorithms with user-friendly interfaces, the project facilitates widespread adoption and usage, thereby safeguarding digital assets and privacy.

Future Directions

Moving forward, the project aims to explore enhancements in model accuracy and efficiency through advanced feature engineering and algorithmic optimizations.

Conclusion:

The project has effectively accomplished its aim to create and implement a cloud-based API for static PE malware detection. This endeavor underscores the efficacy of leveraging machine learning techniques for malware classification and underscores the utility of cloud platforms such as Amazon Sagemaker and Google Colab in crafting scalable, user-centric applications.

Demo Video :

https://www.youtube.com/watch?v=uKgZ4xjgGDw