LONDON METROPOLITAN UNIVERSITY

Containerization and Parameterization of Machine Learning Code

Viktor Sowinski-Mydlarz

Abstract:

This presentation describes the methods for containerizing the Python threat detection Machine Learning application. Containerizing means wrapping an application with necessary libraries, dependencies and config files needed to run effectively on different platforms. We are using Docker technology for that purpose. Our architecture involves 5 docker containers (software packages). They are created using Docker Compose tool. The structure of tasks executable by means of Python code is discussed. These tasks belong to Initial Model Training, Streaming and Updating categories.

The application configuration (16 parameters) was extracted from Python code and placed in a configuration file in an open data interchange format that uses human-readable text. The configuration file is opened by Airflow workflow management platform, which passes the variables into application code. Automatic generation of configuration files from ontology model scenarios is also discussed as well as streaming the packets from the sensor server and analysing them in batches of 1000. The configuration file is generated automatically by parsing an ontology OWL file and adding Machine Learning parameters. We also briefly explain incremental learning by updating and replacing the model automatically, based on evaluation of its performance using data streaming. The future work will involve correlating data with another stream (e.g., windows log files). Finally, we are going to analyse packet payloads for advanced threat detection.

A DevOps practice (software development and the IT operations) involves containerization, making the implementation of DevOps smoother and easier.