How to Train Your Dragon and Its Friends: AI on the Edge with Eclipse Kura

Pierantonio Merlino
Mattia Dal Ben
Eurotech
OUTLINE

• Who are we?
• Why AI on the Edge?
• Know your dragon: Eclipse Kura AI APIs
• What problem are we going to solve today?
• Introduction to Anomaly Detection
• Tutorial
• Upcoming features
Pierantonio Merlino was born in Udine, Italy, in 1980. He graduated in Electronic Engineering (summa cum laude) from the University of Udine, Italy, in 2005 and he received a Ph.D. degree at the same University in 2009. After several years in hardware research, he switched to software development. He joined Eurotech in 2014 and started working on ESF. He is Eclipse Kura committer since 2015.

Mattia Dal Ben graduated in Electronic Engineering at the University of Udine, Italy. He is currently working as Software Engineer at Eurotech in the ESF team. He worked on multiple deep learning projects concerning image processing and anomaly detection.
WHY AI ON THE EDGE?

• Edge AI is the deployment of AI applications in devices close to where the data are located and collected.

• It puts together the best of two worlds:
  • Lower latency, lower network requirements, increased efficiency, data privacy and security, reliability and resilience, …
  • Increase operational efficiency and safety across many industries using AI

Manufacturing

Healthcare

Logistics

Smart cities
EDGE AI CHALLENGES

- How to collect data from heterogeneous sources?
- How to make the data available to the user?
- How to deploy the AI model on the edge device?
- How to protect the AI model?
- How to manage the fleet of deployed devices?
ECLIPSE KURA AND NVIDIA TRITON™: SIMPLYFING AI ON THE EDGE

- Modularity
- Ready-to-use field protocols
- Eclipse Kapua and IoT cloud platforms integration
- Device management
- Nvidia Triton Inference Server™ integration NEW!
Know your dragon: Eclipse Kura AI APIs
KNOW YOUR DRAGON: ECLIPSE
KURA AI APIS

• Kura 5.1.0 introduced a new set of APIs for managing Inference Engines [1,2]
• An Inference Engine is a library or a service that accepts multiple files describing an AI and ML models and allow to perform inference on data.
• The first implementation is based on the Nvidia™ Triton Inference Server [3, 4]
The Nvidia™ Triton Server is an open-source inference service software that enables the user to deploy trained AI models from any framework on GPU or CPU infrastructure.

- It supports all major frameworks like TensorFlow, TensorRT, PyTorch, ONNX Runtime and even custom framework backend.
- It is provided in native and container fashion
- GPU and CPU accelerated
Kura provides three components for exposing the Triton Server functionality [5, 6]:

- TritonServerRemoteService
- TritonServerNativeService
- TritonServerContainerService

Your AI models are important: let’s encrypt them!

Your AI models are important: let’s encrypt them!
• The AI Wire Component can power up your Wire Graph!
WHAT PROBLEM ARE WE GOING TO SOLVE TODAY?

• Create a deep learning anomaly detector from scratch, leveraging the entire Eclipse Kura ecosystem:
  • Data collection
  • Model building and training
  • Model Deployment
Introduction to Anomaly Detection
WHAT IS AN ANOMALY?

• A pattern in data that do not conform to a well-defined notion of normal behavior [7].
• A data point which differs significantly from other data points.
• An observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism [8].
WHY ANOMALY DETECTION?

• The importance of anomaly detection is due to the fact that anomalies in data translate to significant (and often critical) actionable information in a wide variety of application domains [7].

• Finding anomalies is useful in several domains as cyber security, industry, IoT, robotics, etc.

• Anomaly detection can be applied for intrusion and fraud detection, fault detection, etc.
HOW TO DETECT ANOMALIES?

• Statistical analysis:
  • Threshold, proximity and deviation
• Machine Learning algorithms:
  • Supervised
    • Decision trees, XGBoost, ...
  • Semi-supervised
    • Autoencoders, ...
  • Unsupervised
    • GANs, ...

https://xkcd.com/1838/
WHAT IS AN AUTOENCODER?

- An Autoencoders is a specific semi-supervised (or self-supervised) ML algorithm used in several applications.
- An Autoencoder tries to reconstruct the input at the output.
- It consists of an Encoder and a Decoder.
- The Encoder is a NN that maps the input to a lower-dimensional space (code).
- The Decoder is a NN that maps encoded data back to the input.
- The algorithm is trained to have a small reconstruction error.
- Anomalies typically have high reconstruction error.

![Diagram of Autoencoder](image)

- Input: X
- Output: X'
- \( X - X' \) Reconstruction Error
Tutorial
https://colab.research.google.com/github/mattdibi/eclipsecon-edgeAI-talk/blob/master/notebook/AD-EdgeAI.ipynb
UPCOMING FEATURES

• Support for more ML backends:
  • Intel™ OpenVINO™
  • Eclipse Deeplearning4j

• Contributions are welcome!
Questions?
Interested in the cover image?

Look here!

https://labs.openai.com/s/kDAkiocHXgKDY0j3utSLFPPlj
REFERENCES

[1] https://github.com/eclipse/kura/blob/KURA_5.1.0_RELEASE/kura/distrib/RELEASE_NOTES.txt