LinkEdge: Open-sourced MLOps Integration with IoT Edge

Savidu Dias (Solita)  
savidu.dias@solita.fi

Ella Peltonen (University of Oulu)  
ella.peltonen@oulu.fi
Introduction
Project Aim

The aim of the project is to research, design, and develop a platform that enables seamless integration of MLOps practices with edge devices.
eSAAM 2023
on Cloud to Edge Continuum

Design and Architecture
Solution
Implementation

• The solution consists of two command-line tools.
  • Inference Toolkit - packaging models with inference API into a Docker container (used in the CI pipeline)
  • Edge Manager - setting up and managing infrastructure components of an edge device (used in the edge device)
Implementation - Inference Toolkit/API
Evaluation
Finding a dataset

- **Predictive maintenance** is a very valid use case for ML applications at the edge, as anomalies must be identified as quickly as possible.
- The **operating environment** must consist of hardware resources capable of running ML inference.
- Running inference on the **cloud** is not a viable option.
Simulation Environment

Dataset

• Open dataset aimed to detect component failures in an air pressure system (APS) of trucks.
• 76,000 samples, each containing 171 attributes.
• Positive class - component failures for a specific component of the APS.
• Negative class - component failures not related to the APS.
Simulation Environment

Presents an **ideal case** to develop a validation scenario:

- Air pressure system failures **must be identified immediately** as failure to do so may result in severe damage.
- Computer systems in **modern trucks** come equipped with hardware **capable of running ML inference**.
- Trucks operating on the road **may not always have internet access**.
Simulation Environment

Prediction Model

- **Training set with 60,000 examples:**
  - 59,000 belonging to the negative class.
  - 1,000 belonging to the positive class.

- **Test set with 16,000 examples:**
  - 15,625 belong to a negative class.
  - 375 to a positive class.

MLOps Pipeline

- **AWS Sagemaker Pipelines**
Results

Inference Load Testing

![Chart showing response time comparison between Solution and AWS over requests per second.](chart.png)
Results

CPU Usage

[Graph showing CPU usage over time for different solutions and AWS services, with lines indicating different request rates.]
Discussion

• A similar trend could be observed in LinkEdge and AWS Greengrass, although Greengrass performed slightly better in comparison.
  • Can be attributed to the fact that the Greengrass application does not run in a containerised environment.

• Although using Greengrass provides slightly better performance, it comes at the cost of setting up OS and application-specific libraries on each device individually.

• Running inference in containerised environments has the benefit of having the flexibility to work with device-specific models more efficiently.

• A single-edge device running on LinkEdge can handle requests of up to 150 per second, which is sufficient in most cases that require ML inference at the edge.
Conclusion

- The main **objective** addressed is how can a platform that enables the **integration** of MLOps practices with edge devices to be developed using **state-of-the-art** tools and methods.

- **LinkEdge**, is evaluated to have a **performance** that matches that of **existing tools and services**.

- **LinkEdge** offers end users an **open-source** tool to set up Edge-MLOps infrastructure with **flexibility** without relying on **third-party software**.
Thank you!