Driving Mode Detection

From Stream Analyze

Introduction

A manufacturer of mobile industrial short distance transportation vehicles wanted to implement edge AI solutions for detecting if the vehicle was carrying a load. The aim was to be able to monitor vehicle utilization across a huge fleet of vehicles globally to optimize routes. The result from the analysis was also to be shared with the customers-operators of the vehicles to enable joint development of optimized intelligent transportation solutions.

Vehicles produce a large amount of data on their CAN bus and even though all vehicles had mobile connectivity, sending such a large volume of data would quickly use up all of the available data bandwidth. Therefore an ML system is needed to classify the driving mode of the vehicle on-device which can then be sent to the cloud.

Challenge

The development of a driving mode analysis model is a complex challenge. To provide robust classification, a random forest classifier model was needed in conjunction with a Neural Network. The neural network acted as the feature extractor which then fed into the random forest classifier. The output of the random forest classifier was then adjusted using a probabilistic rule model, which mitigated un-probable driving mode transitions inferred by the random forest classifier (e.g. a state transition from ‘no load’ to ‘dropping off a load’).

Executing this model on edge devices was a challenge given due to the complexity of a cascade of models and the tight constraints of the platform. Further, as the converted model executes in the edge devices based on high volume CAN bus data streams, the system needed to be efficient.

MLOps Solutions

To address the challenges above, a complete solution was developed using SA Engine, an end-to-end real time streaming analytics platform for edge devices and microcontrollers. The platform was chosen for this particular application as it supported the broad set of machine learning models needed for this application and provided support for MLOps tools which enabled the analysts to quickly test and deploy new versions of their models onto the vehicles.

In the lab, the neural network and random forest models were implemented using Python and then converted and integrated into the SA Engine pipeline through the SA Engine model conversion tools. Once converted, the models could be executed efficiently in the resource constrained edge devices and could be automated through the SA Engine CI/CD pipeline.

The analysts used the generic CAN bus wrapper of SA Engine to build a compact and unambiguous CAN bus data model that extracted data streams from the CAN bus to the prediction model with minimal delay. This CAN bus data model was bundled and deployed together with the prediction model.

The result of the prediction model was then streamed using SA Engine's communication protocol over 3G/4G to an SA Engine cloud instance for storage and further processing. Selected result streams were forwarded to other cloud-based infrastructure using SA Engine's built-in interfaces for Kafka.

Looking Forward to the Future

The volume of data created by sensors on edge devices is predicted to grow at an unprecedented pace for the next decade and beyond. It is not possible to send all this data to the cloud due to technical and commercial limitations of wireless connectivity solutions. Furthermore, in this project, a combination of several techniques and model types was needed to produce reliable results.

Additional Resources

Stream Analyze Studio
Statistical analysis of large datasets

Forum Questions

- How do you think the data pressures discussed in this case study will impact the needs for edge processing, and in turn the technologies and methodologies needed to efficiently develop and scale edge solutions?
- How do you envision the future of developing real world analysis and AI solutions? Could the selection between different classifier methods (statistical models, Neural Networks, or other ML models) be facilitated?
- What model deployment challenges were faced and how were they dealt with? Were there other MLOps considerations?