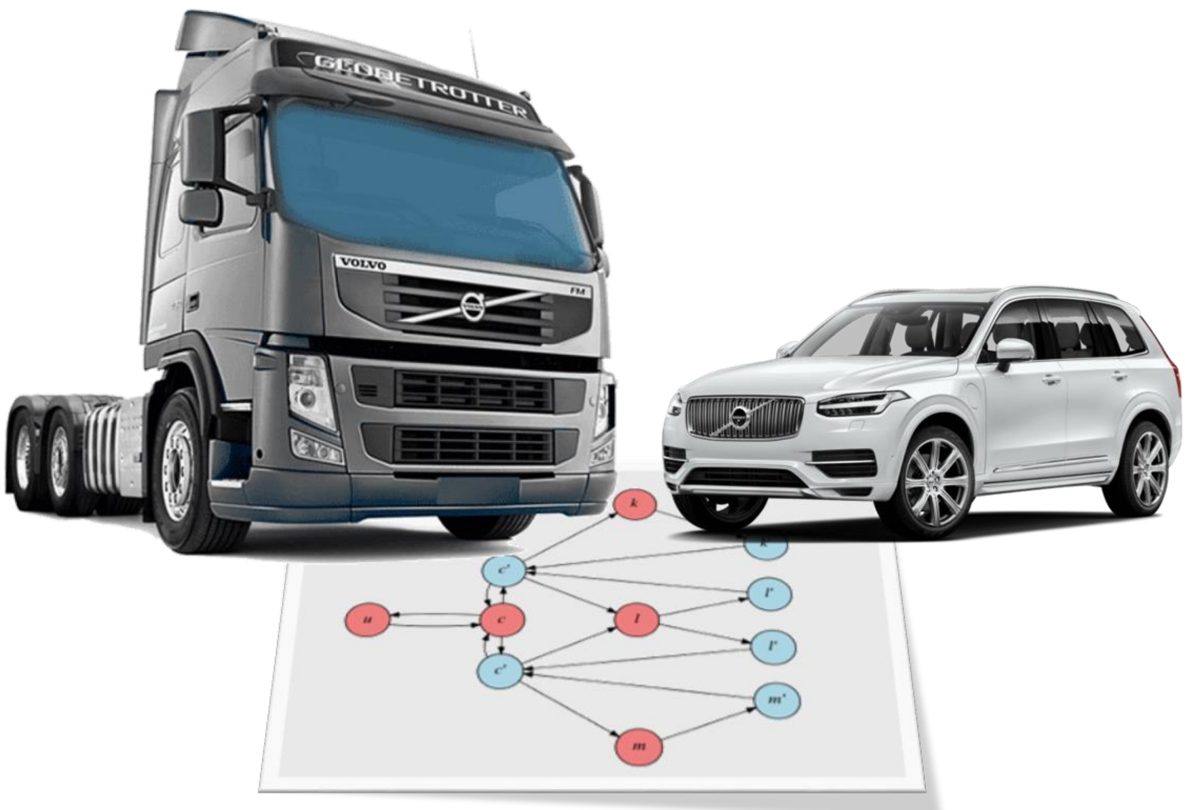
BADA – On-board Off-board Distributed Data Analytics

Public report



Project within FFI – Big Automotive Data Analytics (BADA)
Author Max Petersson - Volvo Car Corporation, et al.
Date 2020-01-24

Content

1. Summary

In the automotive industry, understanding and mapping of customer needs into functional requirements is usually done using engineering knowledge obtained through subjective understanding of the manufacturer. Data analysis of real customer vehicles enables objective understanding of customer behavior, vehicle usage and insights into the driving environment. The current approaches for big automotive data collection and analysis of customer vehicles entails challenges with data transfer, latency, storage and management, data analysis algorithms and methods. Nevertheless, the integrity and security aspects also need to be addressed in order to meet the legal requirements and to establish a trustworthy relationship with the users and fleet owners. The current approaches require the big data computing infrastructure to be either in the cloud or on premises of the automotive manufacturer. Currently it is challenging to centrally store and process the entire collection of vehicle data either on premises or cloud. However, it is more efficient and safer to complement the off-board big data analytics with flexible on-board data analysis capabilities. In addition, and advantage of the on-board data analysis is that sensitive raw data of the customer need not be transmitted to the vehicle manufacturer. Thus there is a need for research and development of a flexible on-board and off-board data processing and analysis platform in the automotive industry.

2. Sammanfattning på svenska

I fordonsindustrin görs analyser och kartläggningar av kundbehov, som sedan blir funktionella krav, vanligtvis med ingenjörskunskap erhållen genom subjektiv förståelse av tillverkaren. Dataanalys av verkliga kundfordon möjliggör objektiv analys gällande kundbeteende, fordonsanvändning och insikt i körmiljön. De nuvarande metoderna för datainsamling inom bilindustrin och analys av kundfordon innebär utmaningar med dataöverföring, latens, lagring och hantering, algoritmer och metoder för dataanalys. Speciellt när det gäller Big Data. Trots detta måste även integritets- och säkerhetsaspekterna tas upp för att uppfylla de lagliga kraven och för att upprätta ett pålitligt förhållande till användare och flottägare. De nuvarande tillvägagångssätten kräver att databehandlingen i Big Data ramverk är antingen i moln eller lokalt hos bilföretagen. För närvarande är det utmanande att centralt lagra och bearbeta hela insamlingen av fordonsdata antingen lokalt eller i moln. Det är emellertid mer effektivt och säkert att komplettera analysen utanför systemet med flexibla dataanalysfunktioner redan i fordonen. Dessutom, och fördelen med dataanalysen ombord är att känslig rådata från kunden inte behöver överföras till fordonstillverkaren. Därför finns det ett behov av forskning och utveckling av en flexibel distribuerad databehandlings- och analysplattform i fordonsindustrin som effektivt fördelar analysen mellan lokalt i fordon och i moln.

Målet för projektet var att utveckla metoder och algoritmer som möjliggör analyser på kundbeteende i riktig kundmiljö. Projektet grundar sig på användarfall som har förfinats

genom en iterativ process och som agerar som kravställning av ramverket och har validerats i flera olika testbänkar. De problemställningar som uppkom när systemet designades blev till grund för mer forskning och jämförelser.

Projektet delades upp i arbetspaket som fokuserade på olika områden. De tre huvudsakliga var metod och algoritmutveckling, dimensionering och implementering samt implementering i en testbänk för demonstration. Arbetspaketen fördelades mellan parterna men ett iterativt samarbete mellan arbetspaket och parter genomfördes genom hela projektets gång.

Metodutveckling och algoritmutveckling
Då man kommer att vilja kunna strömma både rådata och svar på lokala analysfrågor så behövs ett effektivt sätt att hantera dataströmmar, speciellt då det gäller högdensitiva dataströmmar (högt informationsvärde per tidsenhet) så som LiDAR-data. Berörande detta så forskades det omkring och togs fram ett förslag på ett ramverk/arbetsmetod som baserade sig på effektiv komprimering och analys av strömmande data. Komprimeringsmetoden implementerades inte i det ramverk som utvecklades i projektet men har stor potential.

En annan forskningsfråga som påbörjades var hur man effektivt kan välja ut fordon ur en större flotta för frågorna och ändå uppnå resultat med statistisk signifikans. Vissa preliminära resultat finns på denna forskning men kräver mer forskning för att nå ett fullgott resultat.

Dimensionering och utveckling
Detta arbete började med att göra en omvärldsanalys och jämförelse gällande ramverk som hanterar strömmande data, för att kunna välja bästa möjliga spår. Detta arbete slutade med att ett eget ramverk utvecklades som bygger på Erlang för meddelandehanteringen och en analysdel baserad på Python. Erlang är framtaget av Erikson för att hantera meddelanden mellan enheter i ett telefonnätverk. Python är det mest populära språket inom dataanalys och har ett brett utvecklat analysbibliotek.

Ramverket implementerades i en nätverksmiljö där det simuleras fordon som spelar upp verklig fordonsdata men med ansats att kunna ansluta verkliga fordon. Användarinteraktionen baseras på Jupiter Notebook som är välkänd inom dataanalys men kan lätt portas mot andra språk än Python på grund av den inbyggda modulariteten. Funktionaliteten som blev inbyggd baseras på behoven från användarfallen men är mer åt det generiska hållet. Tyngdpunkten ligger på deskriptiv analys men även stöd för användardefinierade funktioner och analysskript finns.

Implementering i en testbänk
För att få in data från fordonen till analysramverket behövs en del processning såsom extrahering och tolkning. I projektet använde vi ett system som utvecklats av Volvo Car Corporation för datainsamling i bilar på väg (WICE) och har förmågan att vara

uppkopplad. Kopplingen mellan detta system och analysramverket utvecklades i projektet och baseras på att få ut en användardefinierad delström av de data som finns tillgänglig i fordonet, ut till klienten i analysramverket, tolkat till fysikaliska värden.

Ramverket och forskningsresultatet demonstrerades vid Volvo Cars Demo Center där en riktig bil utrustades med WICE och en klient installerad på en dator. Bilen kördes på testbanan och verklig strömmande data användes för att visa hur man kan göra analyser centralt (cloud) i systemet bl.a. baserat på geofencing.

3. Background

Knowledge discovery (KD) is one of the key aspects in automotive engineering for designing and dimensioning of vehicle functions, novel products and services. The understanding and translation of customer needs into functional requirements is usually done using engineering knowledge which is obtained under controlled conditions. In the modern age it is very important to quickly capture the rapid changes in customer's functional needs and affected requirements on vehicle functions and systems for a lean product development life cycle. The traditional way for KD about the functional aspects of vehicles is through feedback surveys and targeted customer clinics. These activities are intended to capture the future requirements for customer attributes and set targets for upcoming product design and development projects. The design of survey, social and psychological conditions of the respondent would have a huge impact on the feedback that is gathered. However, insights about customer usage of the vehicle is very subjective and is very difficult to generalize the insights without supporting objective evidence. Modern vehicles are equipped with electronic control units (ECUs) that can make data available from the vehicle sensors and other ECU's. Telematics enable logging and transfer of vehicle data over a communication network which could be further analyzed for product development and understanding the customer usage. Thus, vehicle data logging and analysis from customer vehicles enables the engineers and concept leaders to validate their assumptions from a customer's perspective before huge investments are made. However, deploying vehicle data logging and analysis on a fleet of product vehicles entails issues with data transfer, storage and management, etc. Offline batch processing and analysis of data is currently a methodology for gaining usage and functional insights. However, with increasing amounts of data and the costs involved in data transfer and storage, it is more efficient to process and analyze the data on-board following the distributed processing principles. Hence, in the development of next generation vehicular systems, functions and services, the fleet data is of utmost importance. There have been several developments in this area, providing telematics services that gives an opportunity to connect to the vehicles remotely, assign measurement tasks and further analyze the data. Usually, the telematics systems collect the data based on triggering mechanisms and upload the data in a batch to a remote server or cloud infrastructure. During the early stages of introduction to telematics, the data volumes were comparatively low because of the costs and the primitiveness in the technology development. The collected data could be analyzed with the traditional engineering data analysis tools to obtain insights into limited areas of vehicular functions. With the recent advancements in the telematics technologies it is now possible to collect and transfer large volumes of data from the vehicles in very high resolution and in some cases, it is also possible to stream real time data from a remotely connected vehicle. Today Volvo Car Corporation and Volvo Group Trucks Technology have an estimated need of collecting at least 6.4 GB per hour and 0.4 GB per hour per vehicle, respectively, for these purposes. This would translate into several thousands of petabytes of data that needs to be collected and analyzed per year. With increasing volumes of data, big data analysis technologies have been proven to be efficient for processing and analyzing the collected data. The big data technologies such as Hadoop, Spark etc. are seen as immediate solutions for handling the massive amount of data that is being collected from the connected vehicles. Recent advancements in the domain of big data analysis is the concept of data stream processing where parts of the complete data processing and analysis pipelines could be performed close to the data source before. This minimizes the amount of data that needs to be transferred to a remote cloud infrastructure. Examples of such stream processing platforms are Apache Flink and Apache Storm. The advantage of the data stream processing systems is that the platform supports data operations and computations to be performed in a continuous manner with minimal memory and storage footprint. The streaming paradigm can continually transform the large and fluctuating volumes of raw data into smaller, manageable and insightful streams of data that can then be accessed and analyzed. However, data streaming to a remote or cloud server are hugely dependent on the connectivity. In case of loss of communication network, there is a risk of data loss finally affecting the quality of data analysis results. Thus, stream processing platforms are proven to be effective where data processing and analysis is of utmost importance with high availability of connectivity. The results of the stream processing frameworks could be used for a variety of purposes ranging from efficient and effective data transfer, vehicle functions that requires connectivity and low latency. Examples of such functions include connected convenience, connect safety or even continuous insights and feedback on vehicle condition. All the previously mentioned connected functions share a common challenge in data transmission, bandwidth and expensive data transfer costs. The overall data analysis efficiency can also be improved by balancing the on-board and off-board capabilities depending on the application and weighting of data analysis results. The objective of this project is to develop methods and algorithms that use on-board stream analysis and off-board big data analysis concepts for real data analysis contexts in the car and heavy truck industry. Such application contexts describe customer behavior, vehicle usage patterns, and patterns in logistics and describe the usage environment of the vehicle. The results would have an impact in areas involving verifying the performance of the systems in the consumer environment and for developing the next generation products and services.

4. Purpose, research questions and method

4.1 Methods and algorithms

- Existing batch big data collection and analysis practices are inefficient in terms of data transfer, storage costs and complexity raised due to redundant data collection. Thus, on-board stream processing could be used in combination with the off-board batch processing. The research challenge involves in exploring and identifying the various data stream processing and analysis methods that are suitable for the automotive industry.
- Existing big batch data analysis methods can be efficiently complemented with on-board data stream analysis methods. The research challenges involve query deployment mechanisms that could flexibly balance both on-board and off-board capabilities depending on the availability and the data analysis contexts.
- Furthermore, the robustness of the methods and algorithms both horizontally in the automotive industry (in this case Volvo Car Corporation and Volvo Group Trucks Technology) with respect to advancement in capacity and capabilities needs to be investigated.

4.2 Application of OODIDA approach

Next generation automotive products and services

The future of both car and truck industries would be heavily focusing on connected services that would require both on-board and off-board big data analytics. Thus, there is a need for identifying the differences and overlap between various usage areas in order to make OODIDA as a robust approach. Examples of the next generation products and services include connected safety functions, autonomous drive, truck platooning etc.

Current quality and optimization

Currently there are several analyses and verification (A&V) processes in the automotive industry that are in need of real time customer scenarios. The OODIDA approach would enable these A&V processes to reduce dependency on making assumption of the customer scenarios. The research challenge involves in exploring the possibilities of how the data collection and analysis through OODIDA platform could minimize the product development costs and improve the design effectiveness. In addition, the potential of the

OODIDA approach in gaining insights into the customer behaviour that needs to be explored.

5. Objective

The objective of this project is to develop methods and algorithms that use on-board stream analysis and off-board big data analysis concepts for real data analysis contexts in the car and heavy truck industry. Such application contexts describe customer behavior, vehicle usage patterns, and patterns in logistics and describe the usage environment of the vehicle. The results would have an impact in areas involving verifying the performance of the systems in the consumer environment and for developing the next generation products and services.

6. Results and deliverables

6.1 Use cases

It ended up being four specific use cases that we addressed in the project beside generic analytic capabilities. It has been an iterative process between development or research, and the use cases. The use cases derive both from Volvo Car Corporation and Volvo Group Trucks Technology.

Battery life time estimation
To have a good model of the life time estimation of a battery, the model must be based on several local models (in this case, vehicles) to be able to catch the environmental and usage impact on the battery system. This gives that the use case is a good use case for a distributed analytic framework.
In the beginning the approach was to have a linear regression model that would be updated with data from local linear regression models from the vehicles, but it turned out that it was more complex, and it seems that a more complex machine learning model need to be used if not even a deep learning model. This ended up in completely new research questions, federated machine learning or distributed deep learning, that become out of the scope for this project. Still is the basic concept of the OODIDA framework an enabler for this use case.

Effective handling of LiDAR or geospatial data
This use case addresses the question how we can handle huge amount of data in a streaming fashion in an effective way. LiDAR data are the target because a LiDAR system in a vehicle produces data with a rate of approx. 2 GB/s, but also geospatial data is interesting to handle in the same way. This ended up with the need to have a generic, scalable compression method, on streaming data, that is computation effective (due to the low computation capabilities in a vehicle) and how this can be used in an analytic system.

Geofencing
Often the analytic questions are depending on environmental data like geospatial data. The analytic framework must be able to handle this on a user defined way. By being able to set up a geofence around a point of interest and then running an analysis when the vehicle is inside the fence, is needed for many different use cases like passing a charging station, inside a loading area and more.

Road quality estimation
This use case addresses the durability aspect. The durability of the vehicle is many times depending on the road conditions and how long the vehicle has been driven in different classification of the road condition. The classification can be done in a more local level in the vehicle but also in a more system level based on the local estimations and classifications.

6.2 Methods and algorithm development

6.2.1 Chalmers' Distributed Computing and Systems research group

The work in this task has been carried out by Chalmers' Distributed Computing and Systems research group in the Network and Systems Division of Chalmers' Computer Science and Engineering department (CSE), together with the Big Data team at Volvo Car Corporation, to deliver a set of solid scientific grounds for various methodological aspects of the OODIDA project. In particular, the proposed goals were to deliver and disseminate (through journal articles, conferences and a licentiate thesis) the results of research focusing on:

I. Methods for intra-vehicle multi-sensor data streaming analysis deployable on single-board devices,
II. Methods for both on-board and off-board data analysis, and
III. Exploration of remote deployment of custom queries.

The remainder of this section elaborates on the results with respect to the set goals. The experimental validation for all results of the presented research was majorly performed on the testbed of on-board devices comprised of ODROID-XU4 low-power single-board computers connected by an Ethernet switch presented in Fig. 1.

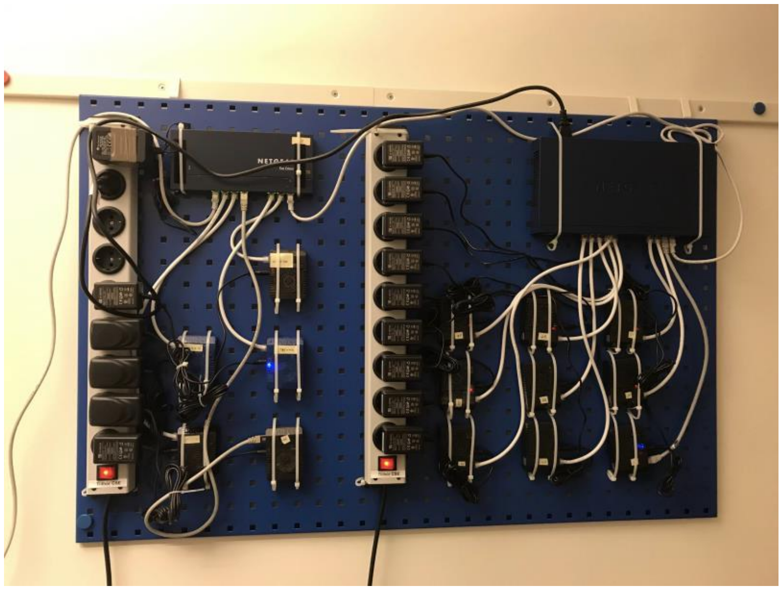


Fig. 1: Testbed consisting of ODROID-XU3 single-board devices connected by an Ethernet switch.

I. Methods for intra-vehicle multi-sensor data streaming analysis deployable on single-board devices

Motivation and challenge:

In a cyber-physical system, data flow and storage from edge devices are hampered when data volumes exceed the internal or external communication bandwidth. This problem is only exacerbated when the number of connected sensors increases. Data volumes can be reduced through well-established compression techniques, but these techniques can be too computationally expensive for on-board hardware or unable to compress data live. The challenge is thus to overcome hardware and implementation limitations of existing solutions to enable high-throughput data streaming pipelines.

Proposed method(s):

We port the technique of Piecewise Linear Approximation (PLA), which trades space against precision by representing some portion of 2-dimensional data by linear segments, into the streaming paradigm to enable low-overhead lossy compression with tuneable errors and compression delays. Furthermore, we introduce streaming-based output protocols to avoid inflating hardly compressible data and reducing the output latency. In particular, we propose a fully independent compression of all channels of a time series to reduce reconstruction latencies, and introduce the concept of singletons to mitigate the data inflation of poorly compressible streams. Finally, we introduce a new PLA construction method combining several known approaches (a best-fit line approximation and maintenance of 2 convex-hulls for checking if the approximation breaks the bound on tolerated error). The new method (*Linear*) is designed to be lightweight and fast as well as aiming at reducing the error due to the approximation.

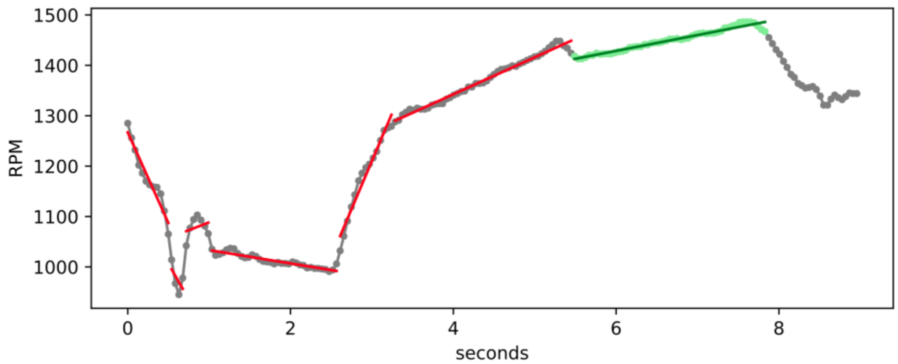


Fig. 2: Illustration of the Linear construction for the Rotations per Minute data collected from a vehicle engine. Input points are processed one by one with a best-fit line approximation (red lines). As soon as the approximation cannot be pursued, a straight segment line is output (green line) and a new approximation segment starts being built. The rightmost grey dots are points not yet processed by the algorithm.

Results (examples/highlights) and impact:

Our work is published in [1]. We have conducted an extensive evaluation of state-of-the-art against our newly introduced "protocols" and our new PLA method. We have evaluated the different solutions on 3 large datasets (19k GPS traces 0.5Gb, 145k LiDAR scans 2.8Gb and 7k 76 days of Speed measurements 2.3 Gb) of real data upon three streaming-oriented performance measures: how well the method was compressing the data (compression), how long we need to wait for receiving outputs (latency) and how far from the truth the approximation was (error). We have shown that we are able to never inflate data, and achieve several orders of magnitude faster reconstruction while keeping a compression level in the same order or better as the best known PLA methods. Moreover, our solutions and especially the newly introduced *Linear* method produces more than twice smaller errors compared with state-of-the-art PLA constructions.

This initial work on PLA compression set the basis for the DRIVEN framework, which leverages these results and expands them to a on-board / off-board framework for data clustering, as described in the following section.

II. Methods for both onboard and off-board data analysis

Motivation and challenge:

A cyber-physical system (CPS) of vehicles connected wirelessly to a data center enables manifold benefits (e.g. live analysis of component and driver behavior, or platooning, etc.) by utilizing the combined processing power of vehicular and server hardware distributed in the CPS, but is bound to many challenges, spanning efficient analysis, efficient communication, security and privacy. A key aspect in this context is the need for solutions that can jointly address several such challenges. When focusing on aspects such as data communication and analysis in such a CPS, a well known challenge is given by the imbalance between the amounts of data sensed and produced by the sensors deployed on the vehicles (a modern vehicle, on the road today, senses more than 20 GB/h of data) and the infrastructures' capacity of gathering that data within small periods of time to the data center and subsequently analyzing it. This scenario requires solutions focusing on efficient data analysis as well as communication aspects.

Proposed method(s):
We propose the DRIVEN framework to tackle the aforementioned challenges for the common problem of gathering and clustering of vehicular data. In a nutshell, the DRIVEN framework jointly addresses the challenges of data gathering, online analysis and leveraging of vehicles'' computational power by:

a. leveraging the aforementioned lossy compression technique, based on Piecewise Linear Approximation (PLA), that significantly reduces the amounts of data to be gathered from vehicles,
b. leveraging state-of-the-art online clustering techniques such as Lisco [15], which overcome the limitations of batch-based ones, and
c. relying on the data streaming paradigm to transparently achieve distributed and parallel deployments.

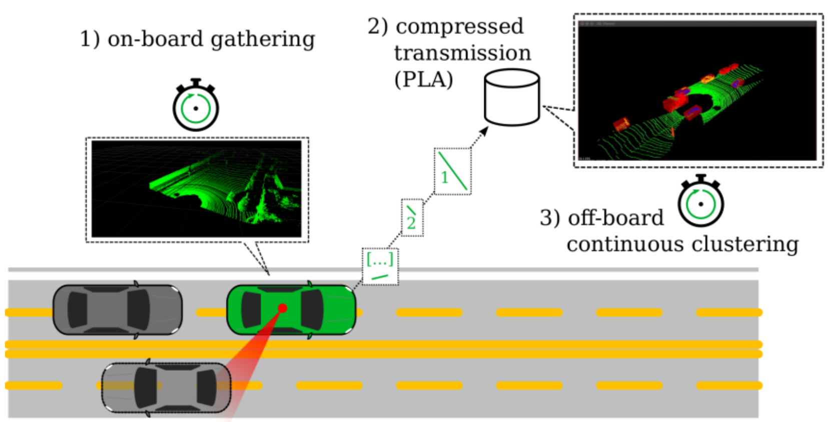
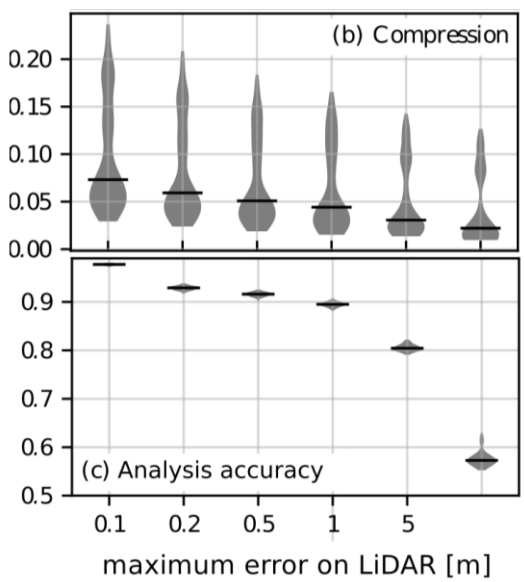


Fig. 3: Example DRIVEN application (green vehicle equipped with rotating LiDAR): 1) Onboard gathering and 2) compression of live LiDAR data, transmission and 3) live offboard clustering.

Results (examples/highlights) and impact:



Our work is published in [2,10]. In the paper, we show with our testbed that the application of our proposed DRIVEN framework to several different use cases, spanning from offboard online clustering of GPS trajectories over CAN to LiDAR data, can yield up to 10 times faster analysis results, while transmitting up to 95% less data and incurring only a 10% precision loss for the analysis. In the figure on the left, we see that the compression ratio and resulting analysis accuracy can be tuned by controlling the maximum error of the PLA technique (here: the maximum deviation of raw LiDAR measurement to the compressed equivalent), allowing to set application-specific parameters for e.g. safety-related applications. Overall, these results underline that smart onboard-offboard processing can yield significant advantages over a traditional send-then-process workflow. These results have been positively received by the community (e.g. Big Automotive Data Preprocessing: A Three Stages Approach, Experimental Study on Machine Learning with Approximation to Data Streams, and A New Pattern Representation Method for Time-series Data), further motivating to continue investigating methods for combining streaming distributed analysis and data compression, and researching other means of sharing analysis workload between the fleet and a central server.

III: Exploration of remote deployment of custom queries

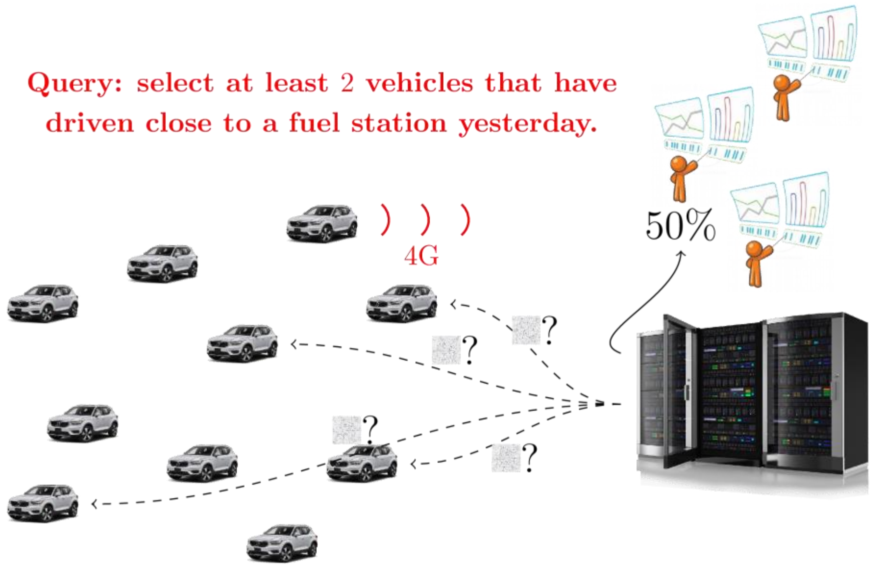


Fig 4.: Illustration of a query-spreading algorithm: analysts at the data center push the query upon a small subset of the fleet using cellular connectivity. The queried subset of vehicles compute locally whether or not they fulfil the query's requirements and send their answer (yes/no) back to the datacenter which can then obtain an estimation of the fraction of vehicles fulfilling the query as well as a number of positive matches available for further analysis.

Motivation and challenge

When analysts issue queries to vehicles that require data satisfying certain criteria (e.g. those driving above a certain speed or in a certain area), instead of forcing all vehicles in a fleet to send all their raw data (and thus congesting the communication and computing infrastructure), the vehicles themselves can check and report the compliance of their local data with the query (e.g. recorded GPS positions or CAN data), hence avoiding a costly data retrieval phase.

In light of limited computational capacity onboard the vehicles, analysts can not afford to engage all vehicles in checking their data's compliance with all queries all the time. The problem to solve thus consists in spreading a set of queries over a vehicular fleet while balancing the time needed to resolve the queries and the computational overhead induced on the vehicles in the network.

Proposed method(s)

We propose a set of asynchronous algorithms for selecting successive subsets of vehicles from a fleet until a certain number of vehicles fulfilling the query condition is found. We furthermore introduce a simulator for testing and evaluating these and compare our solutions to baseline ones such as querying the full fleet and querying new vehicles only upon receiving a negative answer from a queried vehicle that does not fulfill the query condition.

Results (examples/highlights) and impact:
Our work is published in [3]. The balanced algorithms that we have introduced are parametrable and achieve a significantly better trade-off. We have conducted an extensive evaluation of our proposed methods in a realistic simulated environment using real collected data (from a large dataset of GPS positions in Beijing as well as GPS and CAN data from hybrid vehicles collected in Sweden by Volvo Car Corporation), realistic network assumptions and queries (involving geographical and temporal conditions, as well as using CAN-specific signals), and real measured time from running the queries in our ODROID testbed for vehicle computing resources stand-in. In this thorough evaluation, we have shown that our implementations on real data provide on average 36x faster responses (always less than 0.5s to resolve a query) and 70% of computational power is saved with our algorithms compared to baseline solutions.

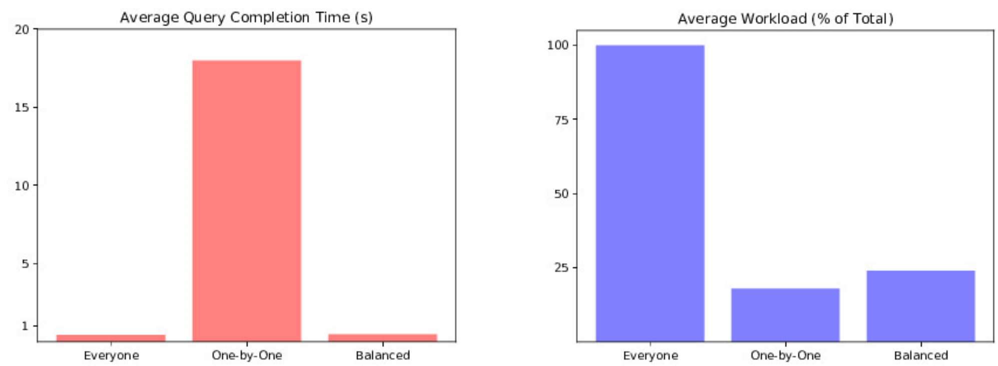


Fig. 5: Summary of the evaluation results comparing baseline approaches Everyone (querying the full fleet to contribute to the resolution of a query) and One-by-One (querying new vehicles only when it is necessary) with our newly introduced asynchronous query spreading strategy Balanced.

The very encouraging results and positive feedback received at the 2019 Intelligent Transportation Systems Conference encourage us to extend our work towards an even more realistic modeling and considering also other possible applications of the spreading strategies. In particular, modeling of a dynamic fleet with vehicles joining and leaving interactively the fleet while processing the queries

Dissemination
The work has been disseminated through top-tier academic channels, including publications in flagship conferences (Symposium on Applied Computing 2019 -ACM SAC, International Conference on Data Engineering 2019 -IEEE ICDE & Intelligent

Transportation Systems Conference 2019 - IEEE ITSC), presented by the industrial PhD student Bastian Havers and the postdoctoral researcher Romaric Duvignau. At the same time, the dissemination will also be carried out via the Ph.D. licentiate and Ph.D. defense of the industrial Ph.D. student, co-supervised both by the industrial partner Volvo Car Corporation as well as by the academic supervisors at Chalmers University of Technology.

When it comes to dissemination to other academic and research institutes as well as computer science students, the DCS research group provides an ecosystem of basic and applied research and a research network in academia and industry internationally, including partners of EU projects, SSF projects, the WASP network, and the Chalmers Areas of Advance (interdisciplinary research platforms in Communication, Energy, Production, Transportation). These imply a variety of natural channels of dissemination, through seminars, workshops and related activities. Moreover, the team engages undergraduate students in projects that associate with the research (through specialized advanced courses such as “DAT300: Data-driven support for cyber-physical systems” (http://www.cse.chalmers.se/edu/course/DAT300/), as well as Masters theses [14]), hence transferring to new engineers knowledge, interdisciplinary expertise and enthusiasm for work on the topics of interest. For example, during study period 2 in the autumn of 2019, the industrial Ph.D. candidate Bastian Havers and the postdoctoral researcher Romaric Duvignau interacted with a team of ambitious masters students on a project on analysis and visualization of live data from a fleet of vehicles. Using a stream-processing engine and modern dashboard visualization, the students created an end-to-end system visualizing in real time the results of off-board queries (Jedvert, Günke, Isgandarli & Muangsiri, *Online Visualization of Live Queries on Vehicular Data*). The students made use of the preprocessed vehicular dataset we have built and their final result opens the door to an integrated visualization for on-board queries as well, such as those deployed remotely in method III.

6.2.2 Chalmers’ Mathematical Sciences

Work in this task has been carried out by Chalmers’ Mathematical Sciences together with Volvo Group Trucks Technology, to investigate the possible future expansions of the initial on-board-offboard system. We propose two separate contributions to future analyses:

- Model development and data analysis of the connection between driver behavior and material fatigue in heavy duty vehicles.

The transportation industry contributes significantly to the economy in both industrial and developing countries. Well-designed heavy duty vehicles are important both to the service life of the vehicle itself as well as the stress on the roads, as road maintenance is expensive while road failure threats traffic safety. Material fatigue, which is the phenomenon where metal structures subjected to repetitive loads fail, and vehicle weight are two of the most important factors to consider when designing a new truck. As a natural next step, we would like to utilize the developed on-board-offboard analytic system is to model the connection between driver behavior and material fatigue in the trucks and analyze its impact on the vehicle design [16].

- Active optimal sampling of client devices in federated learning for unbiased modelling, complexity reduction, and added privacy preservation.

In federated learning, the population of client devices is often large but inconsistent, due to variable network access and limited participation of the devices. In combination with the data often being unbalanced, the estimated models are often biased and thus not a fair representation of the population. We would like to explore the possibility to utilize active machine learning in combination with optimal sampling to produce unbiased models in the federated learning in this project. By sampling a subset of the accessible vehicles at a given time, we both reduce the computational complexity as well as produce unbiased and representative models. Moreover, the sampling procedure contributes to the privacy preservation of the vehicles [17].

6.3 Dimensioning requirements, method extensions and implementation, and feasibility demonstration

Overview

The Fraunhofer-Chalmers Centre developed a platform for distributed real-time data analytics for connected vehicles, which was named after this very project. To avoid confusion, we typeset our system in italics as *OODIDA*. The *OODIDA* platform is a custom solution for the automotive domain. While our initial goal was the development of a virtual prototype, we went far beyond that. As of now, *OODIDA* is a fully working prototype of a distributed system that has been successfully deployed on fully networked devices in various ways (Ethernet, wireless, 4G). We have deployed the client component of this system to computationally restrained edge devices (Raspberry Pi), which approximate the computational power of on-board units (OBUs) in contemporary reference vehicles. On top, the client application of our system has been fully integrated with a test vehicle. We have subsequently demonstrated the system's capabilities of processing live CAN bus data from on-road vehicles. A more detailed overview of the system is given in [12].

The vision of the future *OODIDA* platform is the seamless connection of data analysts and function developers to both the data and the computational resources that are available in a population of vehicles as well as in the cloud. It will facilitate rapid prototyping of new functions and services where future applications are expected to be found in areas such as autonomous driving, safety, maintenance & service, driver-assistance as well as battery and fuel economy. Applications requiring the aggregation and analysis of data from a population of vehicles would benefit from the developed framework, e.g., road condition monitoring. Access to real-time data and computational power of vehicle fleets in the testing, verification and validation phases of product development has the potential to improve product quality and shorten development cycles, with the ultimate effect of reducing the time-to-market for new products.

Architecture

A high-level illustration of *OODIDA* is shown in Fig. 6, with a more detailed overview by means of a component diagram in Fig. 7. The system currently exhibits an *m*-1-*n* architecture where *m* data analysts interact, via one central server, with a fleet of *n* cars. Using one server is a deliberate limitation of the current prototype. If needed, redundancy could be introduced at this point by adding multiple servers, but this is a consideration for a time when this system may target substantial fleets of vehicles.

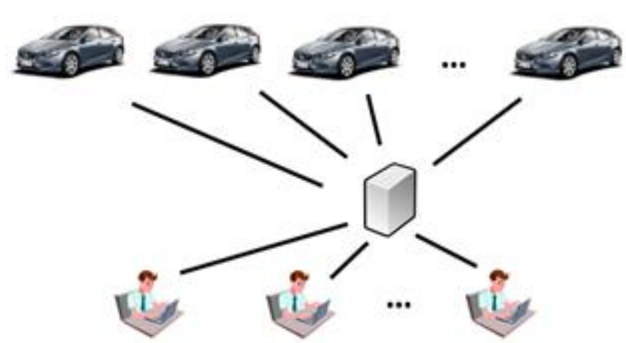


Fig. 6: OODIDA connects m data analysts with n vehicles

A noteworthy aspect of our system is that handles concurrent workloads with great ease. This is facilitated by using the niche programming language Erlang, which was originally developed for the telecommunications domain. Data analysts can, while an assignment is ongoing, issue further assignments that target arbitrary subsets of clients. Meanwhile, all other data analysts can issue multiple assignments as well. The only limitation is in the abilities of the used hardware. Furthermore, our system can be scaled horizontally and vertically. If higher computational demands are placed on vehicles, a more powerful OBU could be used. However, *OODIDA* could easily be modified to address multiple OBUs in one vehicle.

Modularity is a key feature of our system. While the current system has mainly been built with Python (assignment creation and task execution) and Erlang (message-passing infrastructure), it is possible to replace any node at the periphery with ease. Furthermore, it is possible to use different target applications for different use cases. For instance, Python could be used for exploration and prototyping of algorithms whereas a faster language like C or Go could be used for implementing performance-critical algorithms.

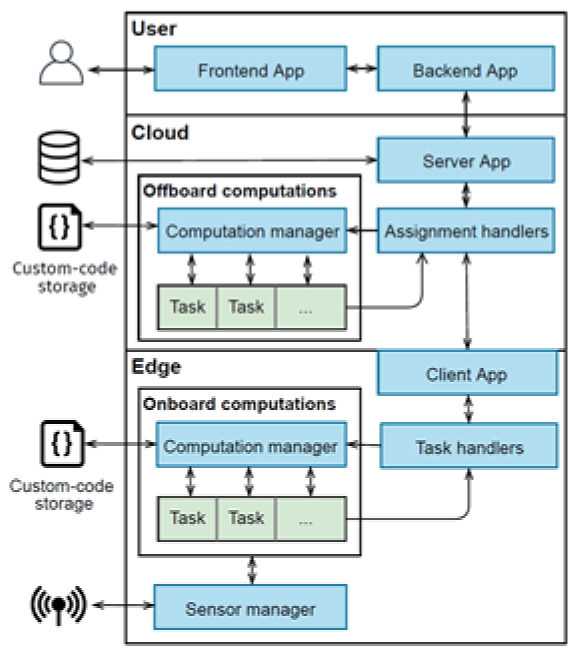


Fig. 7: Simplified overview of the architecture of OODIDA, which highlights its modularity. The backend, server, and client apps as well as the assignment and task handlers have been implemented in Erlang. The user frontend, computation manager and sensor manager have been implemented in Python. Modules or functions that carry out tasks can be implemented in an arbitrary programming language. The sensor manager on edge devices interfaces with a vehicle's CAN bus.

User Interaction

Users interact with *OODIDA* via a Python library. The intended way of using it is via Jupyter notebooks. This makes it possible to integrate generated results into data analytics workflows as users can easily use the data they produce for their subsequent local work.

Users create an assignment specification which includes, among others:

- subset of vehicles
- algorithm with parameter values
- sensors
- number of measurements
- frequency of measurement in Hz
- number of iterations

An assignment specification consists of an on-board (client) and an off-board (server) part. The server takes an assignment as input and turns it into tasks, which are executed by the server as well as all chosen clients. The specified number of iterations, which defaults to 1, enables stream processing with micro batches. This emulates stream processing with a tumbling window, where a fixed number of data points are processed per interval. Of course, the length of the window is arbitrary. It is possible to build algorithms that perform continuous processing of data streams on top of the existing infrastructure.

Active-Code Replacement

The previous description implies any assignment is one of a large number of possible permutations of the specified parameters. However, *OODIDA* is more flexible and allows the user to perform “remote programming.” We refer to our approach as active-code replacement [4, 11]. This feature makes it possible to execute code written by the user on both the client and the server, which is, as shown in Fig. 7, saved in custom-code storage on the cloud and edge nodes. The user has the freedom, with some limitations due to security issues, to deploy a complete Python module to both the cloud and the edge.

As custom code is deployed as an assignment, it is possible to not only use *OODIDA* for rapid prototyping of algorithms. In addition, approaches like A/B testing are made possible by deploying variations of an algorithm to non-overlapping subsets of clients. However, active-code replacement is not intended as an alternative to updating the system with new algorithms. Instead, we view it as a supplementary approach that aims at reducing the turnaround time of the user between conception and exploration. Eventually, such prototyped code should be added to the standard installation of *OODIDA*.

Methods and Use Cases

OODIDA includes a number of hard-coded standard methods that can be applied to finite batches of data or potentially unbounded data streams. In addition, we provide access to standard third-party libraries for statistics and machine learning, such as scikit-learn and Keras. As we are using standard tools, algorithms can be prototyped on a local workstation even without access to our system. Examples of currently available methods include various metrics from descriptive statistics, clustering algorithms, and procedures for anomaly detection.

In a collaboration with Volvo Group Trucks Technology, we implemented two elaborate use cases for distributed analytics that are based on real-world scenarios. The first is geofencing (cf. Fig. 8), with added analytics, e.g., monitoring which vehicles enter and exit a specified area, and providing statistical information on them. Geofences can be intuitively defined by drawing polygons in a GUI. The second use case we worked on is durability monitoring based on road condition data and read-outs from sensor data (cf. Fig. 9). This is helpful for predictive maintenance. Keep in mind, however, that all use cases are implemented as Python code. Visualization is optional but can obviously be very user-friendly.

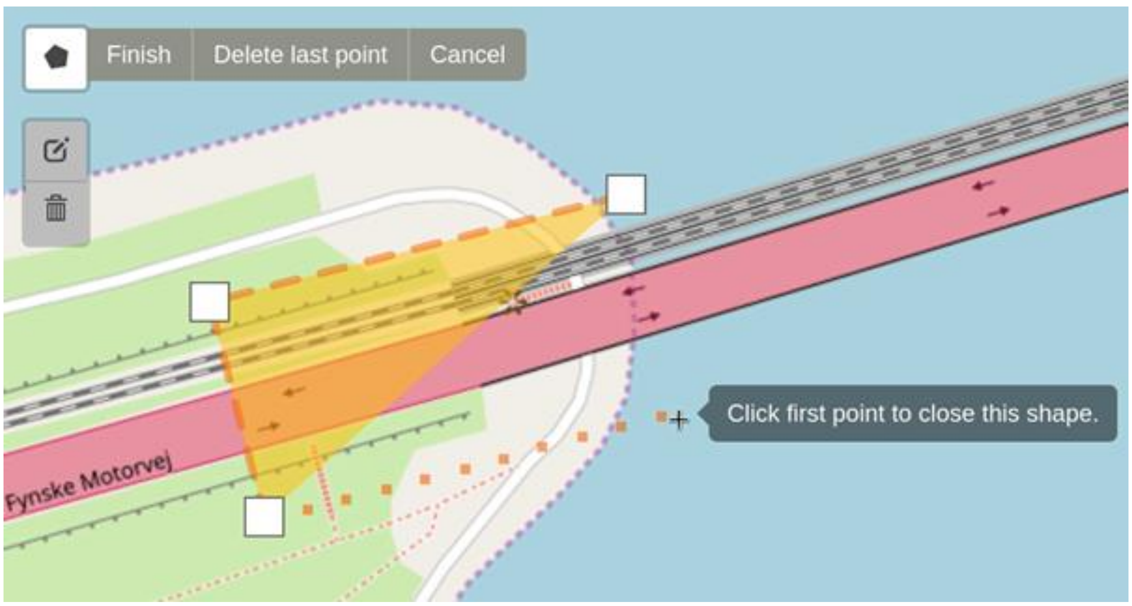


Fig. 8: Geofencing in the user front-end. The user specifies a geofence as an arbitrary polygon. Data analytics are built on top of it, enabling a wide variety of applications. For this use case, the visualization is optional, but potentially more convenient than defining geofences directly in code.

Fig. 9: Visualization of an aspect of predictive maintenance based on street quality and vehicle component status. This is a proof-of-concept. What is shown is a visualization of the quality of a road, based on external data. While this is a mock-up with hard-coded street quality data, data for wear and tear of some components can be accessed via the vehicle's CAN bus. Visualization is optional for this use case.

Scientific Contribution

Our work on *OODIDA* led to the publication of four peer-reviewed conference papers [4, 5, 6, 7], two published conference abstracts [8, 9] (poster presentations), and one Master's thesis [13] in collaboration with Chalmers University of Technology. In addition, we have two more manuscripts in preparation [11, 12]. Some of these contributions relate to the system we built or its features, such as active-code replacement [4, 11]. However, we also used *OODIDA* as a vehicle for research in applied machine learning. We carried out a comparative evaluation of several federated learning algorithms [6, 13] and explored federated learning with deep neural decision forests [5].

6.4 Implementation of on-board and off-board methods and algorithms

In order to implement on-board and off-board analytics, the telematics system WICE, developed by Alkit and used heavily by Volvo Car Corporation, has been utilized as a demonstration and experimentation platform. The WICE system includes an in-vehicle data processing and communication unit which provides access to in-vehicle data sources (e.g. communication buses such as CAN and FlexRay, ISO14229 diagnostic data from ECUs) and supports connectivity with back-end (cloud) computing infrastructures.

In the OODIDA project, the access to vehicular data sources for onboard (and offboard) processing has been greatly simplified through the development of a Signal Broker software component, which provides a publish/subscribe API to in-vehicle sensor signals, relieving the data analysts of the complications of reading e.g. CAN or FlexRay frames and interpreting the data into time series signals. The WICE system also provides a Rapid Prototyping (RP) feature (developed in the earlier FFI project BAuD and extended in the OODIDA project), which allows custom code to be installed and executed on the in-vehicle WICE units, and providing communication of data to the back-end WICE infrastructure, where it can be accessed by off-board analytics components. Specifically, in the OODIDA project, support for using software containers based on Docker in RP concept has been explored.

The WICE RP and Signal Broker mechanisms together provide a powerful platform for realizing joint onboard and offboard data analytics, that has been used in the OODIDA project for Proof-of-Concept implementation and demonstration of key Volvo Car Corporation and Volvo Group Trucks Technology Use Cases.

The OODIDA framework for analytics of data streams from within a Jupyter Notebook (described in section 6.3) has been demonstrated with live data from a vehicle using the WICE Signal Broker approach. For this integration work the WICE Signal Broker serves as a bridge between the demonstrator application and the vehicle's CAN and FlexRay buses. With existing reference vehicles, this type of integration can be, and has been, done in as little as 15 minutes. The proof-of-concept demonstration set-up was realized using an additional computational device that connects to the OBU, e.g., a laptop that executes the client part of our system and also connects to the 4G network. However, in the future, we expect the client part to be executable directly on an OBU with a 4G or 5G module that connects to the *OODIDA* server application. This should also make it possible to fully automate the deployment of the system as the manual steps that are currently necessary would be eliminated.

A geofencing Use Case was demonstrated at the final demonstration event, wherein the position of a car driven on a test track was continuously processed and visualized. More or less any type of data processing of time series signals can be realized this way.

7. Dissemination and publications

7.1 Dissemination

How are the project results planned to be used and disseminated?	Mark with X	Comment
Increase knowledge in the field	X	
Be passed on to other advanced technological development projects	X	
Be passed on to product development projects	X	
Introduced on the market	X	
Used in investigations / regulatory / licensing / political decisions		

Chapters 6.2, 6.3 and 6.4 of this report describe in more detail how the project results have been disseminated within the research consortium and towards the external audience in form of workshops, demonstrations and publications. In the end of the project, a demonstration were held with live streaming data from a car on a test track. Research carried out in the OODIDA project, combined with refined and new use cases from the automotive industry, has led to new requirements and further research needs have been identified. These serve as a basis for future research project applications. Furthermore, industrial partners have started investigating how the OODIDA project results could be implemented into their existing applications, expanding beyond the original OODIDA use cases.

7.2 Publications

Conference papers (peer-reviewed):

1. Duvignau, R., Gulisano, V., Papatriantafilou, M., & Savic, V. (2019, April). Streaming piecewise linear approximation for efficient data management in edge computing. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing* (pp. 593-596). ACM.
2. Havers, B., Duvignau, R., Najdataei, H., Gulisano, V., Koppisetty, A. C., & Papatriantafilou, M. (2019, April). DRIVEN: a framework for efficient Data Retrieval and clusterIng in VEhicular Networks. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)* (pp. 1850-1861). IEEE.
3. Duvignau, R., Havers, B., Gulisano, V., & Papatriantafilou, M. (2019, October). Querying Large Vehicular Networks: How to Balance On-Board Workload and Queries Response Times. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)* (pp. 2604-2611). IEEE.
4. G. Ulm, E. Gustavsson, M. Jirstrand. Active-Code Replacement in the OODIDA Data Analytics Platform. In Proceedings of the 25th International Conference on

Parallel and Distributed Computing (Euro-Par 2019), University of Göttingen, Göttingen, Germany, 26-30 August, 2019. https://arxiv.org/abs/1910.03575

5. A. Sjöberg, E. Gustavsson, A. C. Koppisetty, M. Jirstrand. Federated Learning of Deep Neural Decision Forests. In Proceedings of the Fifth International Conference on Machine Learning, Optimization, and Data Science 2019, Siena, Italy, 10-13 September, 2019.https://doi.org/10.1007/978-3-030-37599-7_58
6. A. Nilsson, S. Smith, G. Ulm, E. Gustavsson, M. Jirstrand. A Performance Evaluation of Federated Learning Algorithms. In Proceedings of the 2nd Workshop on Distributed Infrastructures for Deep Learning (DIDL 2018). Rennes, France. 10 December, 2018. https://doi.org/10.1145/3286490.3286559
7. G. Ulm, E. Gustavsson, M. Jirstrand. Functional Federated Learning in Erlang (ffl-erl). Silva J. (eds) Functional and Constraint Logic Programming. The 26th International Workshop on Functional and Logic Programming (WFLP 2018). Lecture Notes in Computer Science, vol 11285. Springer. https://doi.org/10.1007/978-3-030-16202-3_10

Conference abstracts:

8. G. Ulm, E. Gustavsson, M. Jirstrand. Functional Federated Learning in Erlang. In Proceedings of the 5th Swedish Workshop on Data Science (SweDS 2017), December 12-13, 2017, University of Gothenburg, Gothenburg, Sweden.
9. G. Ulm, E. Gustavsson, and M. Jirstrand. Purely Functional Federated Learning in Erlang. In Proceedings of the 29th Symposium on Implementation and Application of Functional Languages (IFL 2017), August 30 - September 1, 2017, University of Bristol, Bristol, United Kingdom.

Journal articles:

10. Havers, B., Duvignau, R., Najdataei, H., Gulisano, V., Koppisetty, A. C., & Papatriantafilou, M. (2020). DRIVEN: a framework for efficient Data Retrieval and clusterIng in VEhicular Networks. Accepted for publication in: *Future Generations Computing Systems.*

Manuscripts in preparation:

11. G. Ulm, S. Smith, A. Nilsson, E. Gustavsson, M. Jirstrand. Facilitating Rapid Prototyping in the OODIDA Data Analytics Platform via Active-Code Replacement. https://arxiv.org/abs/1903.09477
12. G. Ulm, E. Gustavsson, M. Jirstrand. OODIDA: On-board/Off-board Distributed Data Analytics for Connected Vehicles. https://arxiv.org/abs/1902.00319

Master theses:

13. A. Nilsson and S. Smith. Evaluating the Performance of Federated Learning: A Case Study of Distributed Machine Learning with Erlang. Master thesis,

Chalmers University of Technology, May 2018.
https://hdl.handle.net/20.500.12380/256401

14. E. Nyberg. On-board vehicle data analysis. An evaluation of possibilities using stream processing on anon-board low-powered computer. Master thesis, Chalmers University of Technology, 2018.

7.3 Other references

15. Najdataei, Hannaneh, Yiannis Nikolakopoulos, Vincenzo Gulisano, and Marina Papatriantafilou. "Continuous and parallel lidar point-cloud clustering." In 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS), pp. 671-684. IEEE, 2018.
16. Moaaz & Ghazaly, “A review of the fatigue analysis of heavy duty truck frames”, AJER, 2014.
17. Imberg et al., “Optimal sampling in unbiased active learning”, to appear in AISTATS, 2020.

8. Conclusions and future research

When we began developing this system in 2016, there was no mature data analytics framework available that could easily be modified for the demands of the automotive industry. To our knowledge, at the conclusion of this project in early 2020, this still holds. In addition, as would be expected, there is no open-source data analytics framework for analytics in vehicular networks available that could be used with the vehicles of our project partners. In the meantime, potential commercial alternatives such as NVIDIA’s EGX or Amazon’s AWS IoT Greengrass have been released. Problems we see with those solutions are manifold. The former, for instance, necessitates using relatively costly hardware. Both would furthermore lead to undesirable vendor lock-in. This is a particularly relevant issue for us as *OODIDA* handles sensitive data, which our project partners would prefer to keep on a private cloud. Lastly, Huawei released the first version of KubeEdge at the end of 2018. Their system, which is still an early prototype and, as of now, therefore hard to recommend as a foundation for future work, has some similarities with *OODIDA*, yet it introduces numerous dependencies on other applications, such as Kubernetes.

There are custom solutions, such as stream processing engines (SPEs) that could be executed on OBUs. We considered this as a possibility but found that standard SPEs like Apache Spark are not suited for our scenario due to their comparatively steep performance requirements. The one exception we found was Apache Edgent, which targets resource-constrained devices. Building a client on top of Edgent seemed too limiting. However, an internal fork of *OODIDA* is able to address Edgent for stream processing tasks. The Apache Edgent project has been retired in the meantime. This illustrates one of the risks of wanting to use available off-the-shelf components for internal projects, i.e. the lack of control over development. Furthermore, even if Apache Edgent was still active, it would arguably be difficult if not impossible to get the project owners to make changes that accommodate our needs. As a consequence, it would be necessary to maintain a local fork, which would eventually erode the benefits of using an existing solution. Consequently, we think that our decision to build a new system from the ground up, and according to our requirements, was the right choice.

The *OODIDA* is, together with the related research done in this project, an enabler for resolving the vehicle industry need of accessing and analyse vehicle data with real customers and in real customer environments. As it stated in this report, there is a need for more extended research on some topics and also it's opening up more related research questions.

- **Advanced analytic methods in a distributed environment**
  Some direct need sprung out of this project is how to cope with advanced machine learning like deep learning in a distributed system with vehicle data. There is research done in the academic world regarding methods like this, but applied research and adaptations must be done to be able to use them in a framework like *OODIDA* and with vehicle data.
- **Effectively ask analytic questions in a fleet**
  More research regarding how to select the receivers of the analytic question in a fleet and still have a good statistical significance needs to be done. The work has been started in this project, but needs to be continued. Also, how to implement such methodology in a framework as *OODIDA* needs to be resolved.
- **Deeper implementation in OBU:s**
  The *OODIDA* lacks today the capability to extract and transform vehicle data. This capability and the OBU hardware are quite vendor specific. There is a need to pack the *OODIDA* clients in a way that it seamlessly can interact with the different OBU:s in the vehicle.
- **Usefulness for more than analytics**
  The project has touched upon rapid prototyping and A/B-testing, but no further research has been done regarding this. There also is a need to research around the privacy aspect. One topic is to find a way to measure how much privacy the streaming mechanism and a framework as *OODIDA* gives by default referring to the fact it's minimises the data sets needed for the analytic question.

9. Participating parties and contact persons

Academic partners:
Chalmers tekniska högskola, Computer Science and Engineering
Marina Papatriantafilou (ptrianta@chalmers.se)
Chalmers tekniska högskola, Mathematical Science
Marina Axelson-Fisk (marina.axelson-fisk@chalmers.se)
Fraunhofer Chalmers Research Centre for Industrial Mathematics
Mats Jirstrand (matsj@fcc.chalmers.se)

Industry:
Volvo Group Trucks Technology
Anders Nord (anders.nord@volvo.com)
Volvo Car Corporation
Ashok Koppisetty (ashok.chaitanya.koppisetty@volvocars.com)

Industrial partner:
Alkit Communication AB
Mathias Johanson (mathias@alkit.se)