Big Data Management Towards Impact Assessment of Level 3 Automated Driving Functions

Nisrine Osman

Supervisor: Francesco Bellotti

Submitted to: Dipartimento di Ingegneria Navale, Elettrica, Elettronica e delle Telecomunicazioni of

The University of Genova joint with

The Lebanese University

In Fulfillment

Of the Requirements

For the Degree

DOCTOR OF PHYLOSOPHY

JULY 2021

Dedication

To my family, my source of inspiration.

Acknowledgements

The research leading to these results has received funding from the European Commission Horizon 2020 program under the project L3Pilot, grant agreement number 723051. Responsibility for the information and views set out in this thesis lies entirely with the author. I would like to thank partners within L3Pilot for their cooperation and valuable contribution.

I would like to express my gratitude to my supervisor Prof. Francesco Bellotti who was leading the related tasks in Elios lab (University of Genoa) and to thank all my colleagues.

I would like to thank the coordinator of L3Pilot project Aria Etemad (VW) and to thank Sarah Metzner together with the EICT group for the organization support particularly in annual assemblies, where I learned more about the project through fruitful workshops and presentations prepared by the partners.

Thanks to all the researchers in L3Pilot especially whom I mainly interacted with during this work: Ash Rahman (JLR), Ben Nagy (JLR, former leader of subproject 5), Barbara Metz (WIVW), Erik Svanberg (SAFER), Eduardo Arnold (Warwick Univ.), Hendrik Weber (IKA, leader of subproject 7), Henri Sintonen (VTT), Johannes Hiller (IKA), Luisa Andreone (Centro Ricerche FIAT, leader of subproject 6), Marijke van Weperen (TNO), Guilhermina Torrao (Leeds Univ.), Natalia Neila (JLR, leader of subproject 5), Ricardo Berta (UNIGE), Sami Koskinen (VTT), and Tyron Louw (Leeds Univ.).

Thanks to all my friends for having made my journey more beautiful in Italy.

Thanks to my family for the support during my doctoral study and for the love and encouragement you gave to me all the way.

Abstract

As industrial research in automated driving is rapidly advancing, it is of paramount importance to analyze field data from extensive road tests. This thesis presents a research work done in L3Pilot, the first comprehensive test of automated driving functions (ADFs) on public roads in Europe. L3Pilot is now completing the test of ADFs in vehicles by 13 companies. The tested functions are mainly of Society of Automotive Engineers (SAE) automation level 3, some of level 4. The overall collaboration among several organizations led to the design and development of a toolchain aimed at processing and managing experimental data sharable among all the vehicle manufacturers to answer a set of 100+ research questions (RQs) about the evaluation of ADFs at various levels, from technical system functioning to overall impact assessment. The toolchain was designed to support a coherent, robust workflow based on Field opErational teSt supporT Action (FESTA), a well-established reference methodology for automotive piloting. Key challenges included ensuring methodological soundness and data validity while protecting the vehicle manufacturers' intellectual property. Through this toolchain, the project set up what could become a reference architecture for managing research data in automated vehicle tests. In the first step of the workflow, the methodology partners captured the quantitative requirements of each RQ in terms of the relevant data needed from the tests. L3Pilot did not intend to share the original vehicular signal timeseries, both for confidentiality reasons and for the enormous amount of data that would have been shared. As the factual basis for quantitatively answering the RQs, a set of performance indicators (PIs) was defined. The source vehicular signals were translated from their proprietary format into the common data format (CDF), which was defined by L3Pilot to support efficient processing through multiple partners' tools, and data quality checking. The subsequent performance indicator (PI) computation step consists in synthesizing the vehicular time series into statistical syntheses to be stored in the project-shared database, namely the Consolidated Database (CDB). Computation of the PIs is segmented based on experimental condition, road type and driving scenarios, as required to answer the RQs. The supported analysis concerns both objective data, from vehicular sensors, and subjective data from user (test drivers and passengers) questionnaires. The overall L3Pilot toolchain allowed setting up a data management process involving several partners (vehicle manufacturers, research institutions, suppliers, and developers), with different perspectives and requirements. The system was deployed and used by all the relevant partners in the pilot sites. The experience highlights the importance of the reference methodology to theoretically inform and coherently manage all the steps of the project and the need for effective and efficient tools, to support the everyday work of all the involved research teams, from vehicle manufacturers to data analysts.

Keywords: Automated Driving, Consolidated Data Base, Data Sharing and management, Confidentiality

Contents

List of Figures

List of Tables

1. Introduction

Automated driving (AD) is a hot topic for research both at academic and industrial level [1–5]. Particularly, there is a growing effort towards validating automated driving functions (ADFs) [6–10]. In this context, analyzing field data from extensive road tests is of paramount importance to guarantee effectiveness of AD and assess its impact.

The L3Pilot project, which is introduced in section 1.1 and in which context this thesis has been made, aimed to analyze data from extensive road tests to guarantee effectiveness of AD and assess its impact. It thereby developed a toolchain to support an efficient workflow for AD data processing and sharing among different vehicle owners (i.e., car manufacturers and suppliers who prepared prototype vehicles used in the pilot sites).

Vehicle owners involved in L3Pilot performed the driving test on different road types in different countries, logging a huge amount of data. The researchers in L3Pilot generated a list of 100+ research questions (RQs) to assess the impact of ADFs at various levels: technical system functioning, user acceptance, traffic and mobility and societal level [11]. These questions had to be quantitatively answered based on the data collected in the project.

The FESTA Methodology [12] (presented in chapter 3) details all the steps for pilot management. However, to the best of our knowledge, specific tools that support daily activities of data management as data logging, synthesis, and querying (and with specific requirements in terms of confidentiality and data quality check) are not available in the state of the art. The idea was, thus, to develop an ad-hoc toolchain.

There was a need to support homogeneous workflow by many project partners, such as vehicle owners, research institutions, suppliers, and developers, with different perspectives and requirements. Data confidentiality was at stake and the proprietary data were of different types: objective (vehicular data), and subjective (user data). Overall, there was the need to process the source automotive data to obtain quantitative information that answer RQs considering several factors such as different experimental conditions, driving context i.e., road types (motorway, urban, parking), and driving scenarios.

To face these challenges, L3Pilot designed and deployed a data toolchain able to support the different types of actors to perform efficiently their tasks.

To allow homogeneous and efficient processing of data from heterogeneous proprietary sources, L3Pilot defined a common data format (CDF), onto which all the source signals were mapped. The Common Data Format (CDF) has been released open source, and this is a fundamental resource for the research community in ADFs study and assessment [13].

L3Pilot also defined performance indicators, which are statistical summaries of the signals computed in meaningful intervals of a trip (i.e., the driving scenarios, beside the whole trip). A key novelty introduced by L3Pilot is the use of the consolidated database (CDB), a cloud system sharing all the performance indicators collected through the project's pilot sites. The overall data toolchain was deployed and used by vehicle owners and related partners (for raw data processing, check, and insertion to the CDB). Ultimately, a set of industrial psychology and sociology researchers have used the toolchain on a daily basis, who are responsible for assessing the impact of AD based on the pilot data.

The toolchain has been developed by the partners for L3Pilot project and will be exploited and extended in future collaborative projects, according to the availability of the partners.

Information about the design, deployment and testing of the toolchain has been published in some journal [14,15] and conference [16] articles and in this thesis.

As the University of Genoa is one of the partners who contributed to the development of the toolchain, the thesis will give an overview of the overall system, with specific attention onto those modules to which the candidate mainly contributed.

1.1 The L3 Pilot Project

Automated driving represents a highly challenging technological domain, with a huge amount of research being carried out in the field (e.g., [17–19]). Over the years, many projects paved the way to introduce the AD to the market [20–22]. In this pre-competitive industrial research context, L3 Pilot performs extensive on-road testing of automated driving functions (ADFs) of SAE levels three (L3) and four (L4) [23]. Thereby, it exposes AD to mixed traffic environments: high speed motorway, traffic jams, urban and parking along different road networks. The main objective of the project is to check that the AD performance is consistent, reliable, and predictable. Additionally, it assesses the user acceptance and interaction with AD deployed systems [24,25]. The L3Pilot consortium brings together different partners, including original equipment manufacturers (OEMs), suppliers, research institutes, infrastructure operators, governmental agencies, insurance sectors, and user groups.

OEMs partners are the following:

- Volkswagen AG (coordinator)
- Audi AG
- BMW AG

- Centro Ricerche FIAT
- Ford
- Groupe PSA
- Groupe Renault
- Honda R&D Europe
- Jaguar Land Rover
- Mercedes-Benz AG
- Opel Automobile GmbH
- Toyota Motor Europe
- Volvo Car Corporation.

Research institutes comprise:

- BAST - the Federal Highway Research Institute
- DLR - German Aerospace Center
- ICCS - Institute of Communication and Computer Systems
- IKA - Institute for Automotive Engineering at RWTH Aachen University
- SAFER - Vehicle and Traffic Safety Centre at Chalmers
- SNF - Centre for Applied Research at NHH
- TNO - Netherlands Organization for Applied Scientific Research
- University of Genova
- University of Leeds
- VTT Technical Research Centre of Finland
- WIVW - Würzburg Institute for Traffic Sciences GmbH

- WMG - University of Warwick.

Suppliers are:

- Aptiv
- FEV GmbH
- Veoneer

Small and Medium Enterprises (SMEs):

- ADAS Management Consulting and European Center for Information
- European Center for Information and Communication Technologies GmbH

Insurers:

- AZT Automotive GmbH
- Swiss Reinsurance Company

Authorities:

- RDW - Netherlands Vehicle Authority

User Group:

- FIA - Fédération Internationale de l'Automobile

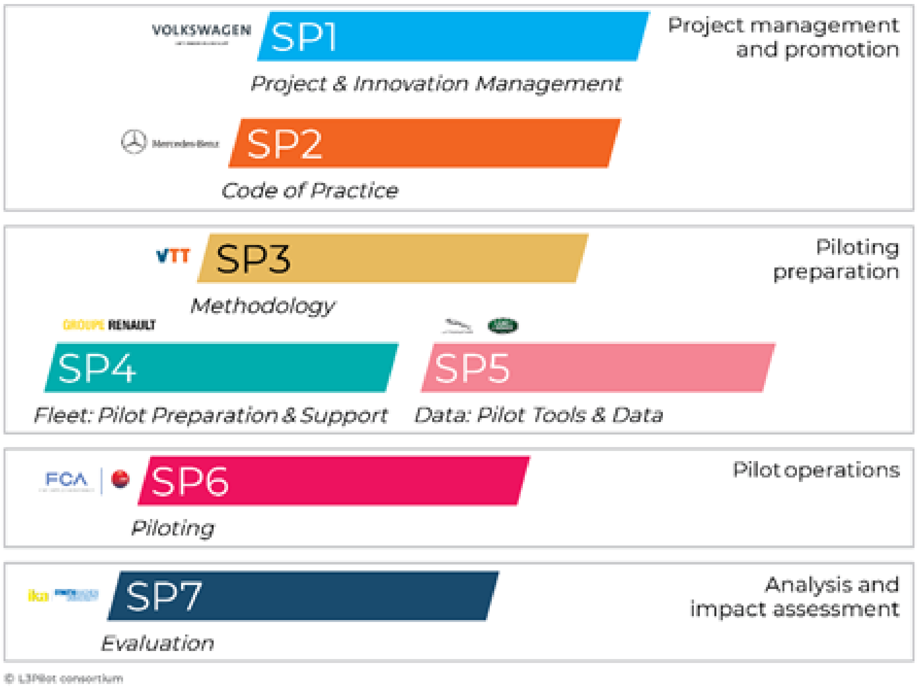

Figure 1-1 L3Pilot structure [25]

Figure 1-1 illustrates the structure of the project with all its sub-projects (SP) [25]. SP3 ("Methodology") defined and managed the project methodology, and defined a list of research questions (RQs) to assess the impact of ADF in four key areas such as: (I) technical performance of the tested L3 ADFs, (II) user acceptance and behavior, (III) impact on traffic and mobility and (iv) societal impacts (see [11]). SP4 ("Pilot preparation and support") and SP6 ("Piloting") developed experimental procedures to collect required data to answer the RQs, and a robust evaluation plan to ensure that reliable and valid results are achieved from the pilot testing [25]. SP7 ("Evaluation") performs the data evaluation and analysis. SP5 ("Pilot tools and data") developed robust methodologies and tools to define and collect required data to answer the RQs. The University of Genova participated in SP5 and SP6.

1.2 Outline of the Thesis

The remainder of the thesis is organized as follows. Chapter 2 analyzes literature to draw the relevant state of the art. Chapter 3 describes the reference methodology targeted and applied in L3Pilot. Chapters 4 and 5 go in details of the developed data toolchain, considering objective (i.e., vehicular) and subjective (i.e., from test users) data. Chapter 6 focuses on the deployment of the toolchain in the project's pilot sites. Chapter 7 draws the final conclusions on the presented work.

2. Literature Review

Computer-aided solutions are crucial in automated driving development [26]. In this context, technological solutions are needed to deal with big data to manage knowledge and support development of effective solutions for mobility [27].

The remainder of this chapter is dedicated to analyzing some key fields related to the L3Pilot research, such as: management and sharing of big data in collaborative projects, methodologies for automotive piloting, and driving scenario representation.

2.1 Big Data in Automotive Projects

Data acquisition and telemetry are key factors for quality and performance in vehicle development and management [28]. The euroFOT project already used small built-in devices with flash storage and GPRS network connection to remotely track and upload data during field operational tests (FOT) [29].Recently, connectivity has been introduced in automotive production series, making vehicles highly mobile nodes in the Internet of Things (IoT) paradigm. In this context, [30] presents the Common Vehicle Information Model as a harmonized data model, allowing a common understanding and generic representation, brand-independent throughout the whole data value and processing chain.

Since the volume of the data collected from vehicles using telematic services can be very high, we need to design scalable and efficient systems and frameworks. [31] explore the opportunities of leveraging Big Automotive Data for knowledge-driven product development, and to present a technological framework for capturing data from connected vehicles and analyzing them online.

Concerning the data format, [32] describes an approach to combine standards specified for the automotive test data management with the widely used Unified Modelling Language (UML). The Here company, purchased by a consortium of vehicle owners, developed the HD Live Map comprising a road centerline model and a lane model, enriched with other attributes. They used the Google's Protocol Buffer data format to describe the schema of their data model [33]. [34] implemented an encoding solution for point cloud Lidar big data in the Hadoop distributed computing environment based on the Google Protocol Buffers framework. The Google Protocol Buffers format is binary, compact, highly versatile (so adaptable to any change in structure), and supported by multiple languages as Python, Java. However, it is not supported by some others as MATLAB.

2.2 Data Sharing in Collaborative Projects

Literature is rich of papers on privacy and risk management in projects. For instance, [35] deal with Risk Assessment in Multi-Disciplinary Engineering Projects, [36] with privacy risks when sharing data on information systems. Furthermore, [37] investigates the validity of sharing privacy-preserving versions of datasets. They propose a Privacy-preserving Federated Data Sharing (PFDS) protocol that each agent can run locally to produce a privacy-preserving version of its original dataset. The PFDS protocol is evaluated on several standard prediction tasks and experimental results demonstrate the potential of sharing privacy-preserving datasets to produce accurate predictors. In addition, [38] provides an extensive review of data analytic applications in road traffic safety, with particular attention to crash risk modelling.

Furthermore, [39] deals with integrating diverse knowledge through boundary spanning processes, with a particular focus on multidisciplinary project teams. The concept of a Project Consortia Knowledge Base (PC-KB) is presented in [40] in an integration framework based on semantic knowledge that facilitates project-level communication as well as access to project data across tool and partner boundaries. Commercial companies (e.g., Amazon, Microsoft, Google) have established efficient cloud ecosystems for data management providing very powerful services, but they rely on proprietary technologies, with very limited interoperability and development opportunities for third parties. However, we could not find in the literature guidelines on how to exploit these cloud technologies to support project partners in processing big data to address quantitative research questions.

2.3 Methodologies for Piloting Projects

In recent years, several field operational tests (FOTs) have been executed to test new Advanced driver-assistance systems (ADAS) involving thousands of drivers (e.g., euroFOT [29,41]). With a view to ensure scientific soundness, the Field opErational teSt supporT Action (FESTA) project developed a methodology for field operational tests (FOTs), with three main focuses: user, vehicle, context [42]. This methodology is described in the FESTA Handbook which has been frequently updated according to the latest lessons learned [12]. The FESTA handbook records lessons learned and provides best practices collected in several European FOTs in the last ten years. L3Pilot decided to adapt this methodology to suit the needs of a real-world piloting of automated driving systems which are at an earlier technology readiness level [43]. As methodologies used to evaluate Field Operational Tests (FOTs) lacks guidance about assessing the impact of automated driving on users' behavior and acceptance. L3Pilot used new methods to collecting data during an automated driving pilot [44]. As an example of such studies, [45] investigated the public acceptance of SAE level 3 automated passenger cars through filled questionnaires among 8,044 car-drivers in seven European countries.

Several collaborative industrial research projects have been conducted in Europe addressing the first levels of automated driving. The AdaptIVe project developed several functionalities providing various levels of driver assistance, such as partial, conditional, and high automation [21,46]. Drive C2X investigated cooperative awareness, which was enabled by periodic message exchange between vehicles and roadside infrastructure [47,48]. The FOT-Net Data project provides hands-on recommendations for sharing data of transport research [49].

2.4 Driving Scenarios

Validating and verifying the correct functioning of automated driving systems is a fundamental challenge. High number of possible traffic scenarios arise from varying environmental conditions and unusual and complex situations [33]. A conventional validation process requires the development of new testing procedures. [50] estimated that testing an automated driving function for highways requires 6.6 billion kilometers of driving to statistically undercut the currently expected distance between two fatal accidents. Thus, [51] presented a generic simulation-based toolchain to identify critical scenarios. PEGASUS project [52] designed 6-Layer Model for describing highway logical scenarios [53],[54]. It defined functional, logical, and concrete scenarios for the simulation-based tests [55], using standards like OpenDRIVE [56] and OpenSCENARIO [57]. However, a concern when developing a scenario framework based on data without incorporating knowledge is that only the scenarios that exist in the dataset are detectable [54]. Nevertheless, we needed to design and implement algorithms to detect driving scenarios along the ego vehicle' trip based on the definitions given in the previous projects.

As a concise outcome of the literature review, we can observe that several industrial research projects have addressed the AD challenges in these years. However, L3Pilot required extensive piloting in several sites with a huge number of vehicles, manufacturers, drivers, and conditions to quantitatively assess various types of impacts of AD functions. The FESTA methodology, detailed in the next chapter, was chosen for quantitatively addressing the RQs defined by L3Pilot. Thus, developing a custom toolchain was crucial to implement and support the daily activities required by the FESTA methodology for L3 ADFs, and for all the project partners.

3. Methodology and Requirements

This chapter overviews the methodology developed by L3Pilot SP3 (“Methodology”), for which efficient application L3Pilot researchers and developers (among them, the candidate) developed the data toolchain, that will be described in the following chapters 4 and 5. The methodology relies on Field opErational teSt supporT Action (FESTA) [12], which details how to deal with the various activities to be performed in an automotive piloting project.

3.1 FESTA Methodology

A fundamental step in managing a complex automotive piloting project, combining human and technological aspects, is given by the definition of the methodology, which shapes all the phases of a FOT project. As several FOTs have been conducted in Europe in recent years, the FESTA methodology has established itself as a reference [42], [58]. The FESTA approach gives general guidance on organizational issues, methodology and procedures, data acquisition and storage and evaluation for FOTs. FESTA covers the whole process of planning, preparing, executing, analyzing, and reporting a FOT. The steps that need to be carried out during a FOT are graphically presented in the form of a V diagram, where a correspondence links the preparation layers on the left-hand side and the evaluation layers on the right-hand side.

Figure 3-1 shows the L3Pilot implementation of the FESTA methodology [11][59]. The left side of the V descends from definition of functions and use cases, down to research questions and hypotheses, performance indicators and measures, data collection tools and pilot site set-up. The bottom of the V is the “use” pillar in the FESTA methodology and denotes “drive” in L3Pilot i.e., the data acquisition during tests. In L3Pilot, the drive phase consists of two parts: pre-tests, that we define as pre-pilots, and tests, that we define as pilots. The pre-tests period takes place before large-scale user tests can safely begin. During pre-testing, e.g., function performance is honed, and data collection processes are tested. Traditionally, in FOT terminology, the pre-testing would be called piloting. Finally, the right-side rises mirroring the left side: data processing, data analysis to answer research questions on technical performance, user acceptance and behavior, impact on traffic and mobility, up to societal impacts. The methodology allows considering, at different levels, the point of view of different stakeholders, such as politics, industry, and research.

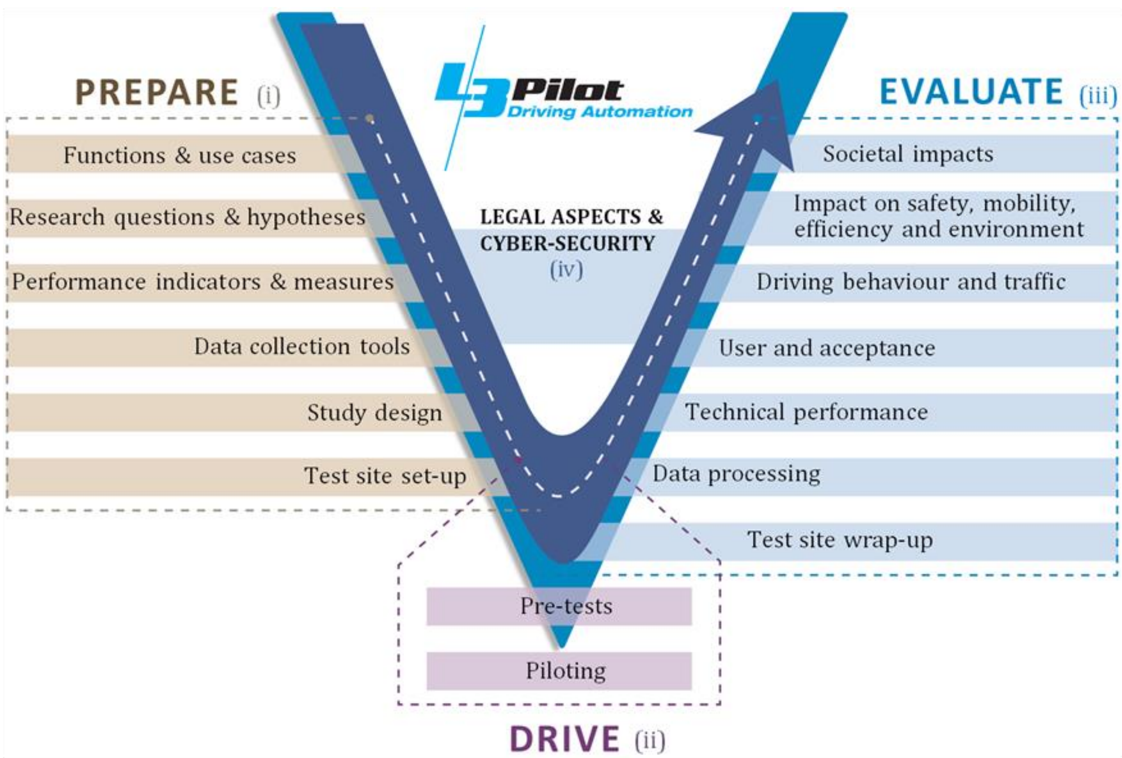

Figure 3-1 L3Pilot Methodology overall structure [11]

Data are in the core of the methodology and pivot on assessment. Consequently, we needed to design an effective and supportive toolchain for collecting the data along testing and assessment phases. To assess the impact of ADF, the l3Pilot, the L3Pilot analysis, based on FESTA, is driven by a set of research questions (RQs) and hypotheses that have been published in the L3Pilot deliverable [11]. Section 3.2 delineates these research questions with an example of logging requirements for a hypothesis.

3.2 Research Questions for L3Pilot

The RQs for all impact areas in the L3Pilot project were generated by L3Pilot SP3 “Methodology” [1] through the top-down approach recommended by the FESTA Handbook [12]. The process began with a review of the descriptions of automated driving functions (ADFs) that were going to be piloted during the project. Therefore, in the early stages, only high-level RQs (Levels 1 and 2 in Table 1 example) were defined, to meet the project objectives.

Table 1 An example of how logging requirements were defined per hypothesis [11]

Item	Example
Evaluation area	Technical &traffic
RQ level 1	What is the impact of the ADF on driving behavior?
RQ level 2	What is the impact of the ADF on driven speed in different scenarios?
RQ level 3	What is the impact of ADF on driven speed in driving scenario X?
Hypothesis	e.g., 1: There is no difference in the driven mean speed for the ADF compared to manual driving. e.g., 2: There is no difference in the standard deviation of speed for the ADF compared to manual driving.
Required performance indicators	Mean speed, standard deviation of speed, max speed, plot (speed/time)
Logging requirements/sensors available	CAN bus of vehicle: Ego speed in x-direction

In such a top-down approach, the generation of the RQs and hypotheses is based typically on theoretical understanding of the mechanisms that influence the different impact areas. The RQs were simply based on literature and the experience of the project members in previous, related work. The generation of the first (higher) level of research questions was structured according to the four L3Pilot evaluation areas. The second stage involved the development of more detailed RQs related to specific components of the higher-level questions, where appropriate. For each RQ, the underlying hypothesis is then made explicit. Table 1 provides an example in the Technical and Traffic area.

In line with the FESTA Handbook, the next steps after generation of the hypotheses concerned the definition of the relevant performance indicators and of the logging needs related to them. Here, we differentiated the subjective and objective data [7]. Questionnaires would collect subjective data across test participants (drivers and possible passengers), and objective data would be collected mostly from the data loggers of the test vehicles, additional cameras installed on them, and, when necessary, from external data sources (e.g., weather information, road type, etc.) [24]. provides an overview of the RQ definition and implementation workflow.

3.3 Vehicular Data

According to the developed methodology, L3Pilot (particularly, SP3, 5 and 7) had to define the source vehicular signals, the derived measures (DMs) (that are intermediate, domain-significant values useful for the computation), and performance indicators (PIs), to be shared from all the pilot sites to provide a quantitative basis for answering the project research questions. The overall analysis led to the definition of a set of signals to be provided by all the vehicle owners [24]. Beside the standard vehicular signals (e.g., speed, acceleration, pedal activity, etc.), source data also come from automated driving sensors, such as cameras, Light Detection, lidars, and radars.

Table 2 shows the four types of PIs defined by L3Pilot as sharable data in the CDB [15]. The table reports only two Datapoint types, as examples since there is one datapoint type for each driving scenario type.

Table 2 An overview of the L3Pilot performance indicators (PIs) types [15]

PI Types	Description	Example of PIs
Trip PI	PIs computed at trip level.	Mean of longitudinal acceleration, percentage of time elapsed per driving scenario type.
Scenario specific Trip PI	PIs computed at trip level but only when a specific driving scenario occurs. Example of driving scenarios, described later, are: Driving in a traffic jam, Lane change.	Mean of duration of sections with speed lower than a threshold
Scenario instance PI	PIs computed for each instance of a driving scenario. The same PIs are computed in each type of scenario.	Mean of time headway, mean position in lane
Datapoint for a Following a lead vehicle scenario Datapoint for Approaching a traffic jam scenario	Datapoint PIs are computed for each instance of a driving scenario. Different types of scenario have a different datapoint structure. Here we report two examples. Datapoints are used as input for the impact assessment by either simulating driving scenarios or constructing artificial scenarios based on statistical analyses of scenarios encountered during piloting.	Mean of relative velocity, Time headway at minimum time to collision Vehicle speed at brake or steering onset, Longitudinal position of object at brake or steering onset

PIs are typically constituted by statistical aggregations (e.g., avg/std/min/max) in significant intervals of a trip. Two PI types are computed at trip level: while Trip_PIs are general indicators synthetizing a trip, ScenarioSpecific_TripPIs are computed aggregating trip segments from a specific scenario only. The other two PI types (namely, ScenarioInstance_PI and Datapoints) are much more specific, as they are computed for each instance of a given driving scenario detected during a trip.

Datapoints are defined to be utilized as feeds to further impact analysis. Not only, in fact, does L3Pilot capture a snapshot of how automated driving technology is and looks like today. It also scales up the detailed findings from log and survey data with various tools, such as macroscopic simulations and transparent stepwise calculations, to estimate higher level impacts of automated driving. Key performance indicators were thus defined by L3Pilot for each driving scenario type in order to enable also simulation studies for impact analysis.

Several research questions required to analyze context data beyond the actual vehicular signals and questionnaire answers. Context data were useful, particularly to segment information so to allow comparisons and more focused analysis. Among context data we highlight:

- Experimental conditions. Different conditions must be considered, such as: baseline, ADF not available, ADF off, ADF on.
- Road types. Tests are performed on various road types, such as: motorways, major urban arterials, other urban roads.
- Driving scenarios. The system must track different types of driving scenarios, that are typical driving situations, such as uninfluenced driving, lane change, lane merge, following a lead vehicle, etc. Scenarios are computed by the L3Pilot data toolchain, processing the vehicular time series.

So, Trip PIs are to be computed in different segments, based on the actual experimental condition (i.e., baseline, ADF off, ADF on) and road type; and Scenario Instance PIs, alike, are to be segmented not only based on the scenario type itself, as per definition, but also considering the different experimental conditions and road types. Other metadata were mandated as well, such as driver type (professional or ordinary), temperature and speed limits to better characterize the context of each PI measurement.

3.4 Subjective Data

A complete assessment of ADF functions requires processing subjective data as well. Subjective data include information collected from participants at the various pilot sites through questionnaires (the complete version of the questionnaires is available in [60]). L3Pilot defined questionnaires for three different types of settings (driving environments and relevant ADF types): motorway and traffic jam, urban and parking.

The first part of the questionnaires includes background information, i.e., sociodemographic questions, vehicle use and purchasing decisions, driving history, in-vehicle system usage, activities while driving, trip choices and mobility patterns. The data collected in the first part were used to create different user groups for the user and acceptance evaluation.

The second part of the questionnaires concerns the ADFs. For example, these questions assess various aspects of participants' initial reactions to using the ADFs.

Finally, the last section is an optional section to evaluate the users' performance during take-over situations in the traffic jam / motorway and urban on-road tests.

When conducting studies across multiple sites, it is essential that any data collection methodologies are applied uniformly. For example, the pilot site questionnaires are administered across all pilot sites, which vary in many respects (e.g., country language), but most relevant here is the interexperimental variability.

To minimize the effect of this variability on the quality of the data in L3Pilot, specific requirements for a common questionnaire management tool at the project level were defined. Basically, these requirements include:

- Conditional questions routing i.e., skip logic. Some questions appear or not depending on previous responses by the test user.
- Multi-language support.
- Different devices. The tool must support inputting data from desktop, tablets, and mobile devices.
- Illustrative multimedia material support.
- Off-line surveying. Beside supporting online questionnaires (filled with tablets, phones, or other devices), the tool should consider an offline (i.e., paper-based) user inputs at the various test sites.
- Data export of responses in different formats as XLS, csv, SPSS, xml, and pdf.
- Questionnaires export to reduce the work in case different installations were needed.
- Information/development support.
- Simplicity of use.
- System integration (e.g., keeping contact with users through e-mails).
- Support for different user roles (administrator, respondents, manager).

- Support for adding jQuery scripts for customized types of questions.

3.5 Confidentiality

Along the project, a major focus was on the confidentiality side, which involved significant discussions in SP 3 (“Methodology”), 5 (“Pilot data and tools”) and 6 (“Pilot”). The consortium decided to not share any private data as trip Ids, user Ids and sociodemographic. It was also decided that data should be pseudonymized before uploading to the CDB and sharing among the partners. The CDB toolchain aggregates data from several sites in such a manner that commercially sensitive information is protected. Beside protecting the privacy of manufacturers, it is also necessary the shared data must describe the impacts of automated driving. The merging of data from different sites corresponds to the fact that the L3Pilot results should not represent the impact of single (OEM)-specific ADFs, but the generic impacts that can be expected once these systems are introduced to the road [14].

A key point concerned the harmonization of the conflicting need for having IDs within the CDB (to avoid duplicates and misses) and for protecting vehicle owner confidentiality and driver privacy. To solve this issue, SP5 (“Pilot tools and data”) decided to apply pseudonymization (or de-identification) to the trip and test participant IDs. Pseudonymization allows identifying entities, but only by the owner of the data (the vehicle owner, in the L3Pilot case), who is the only one knowing the created pseudonymized ID.

The pseudonymized ID is an 8-character string, obtained through a simple procedure, based on SHA-256 hashing. Source information (e.g., driver name, date of birth, trip place, vehicle owner, etc.), integrated with a secret word for “salting”, is processed through a deterministic hash function (e.g., SHA-256), that generates a 64-character identifier [61]. For the purposes of L3Pilot, it was sufficient to extract the left-most 8 characters of the 64-character string to have the driver ID and the trip ID. This way, there are still enough unique combinations with sufficiently low collision probabilities.

In summary, the consortium defined four key requirements for confidentiality, to apply when sharing data to the CBD:

- Stored data should not allow tracing back the original test site. For instance, attention was paid to exclude metadata, such as temperature, speed limit and date, that may lead to reveal the location of the pilot site.
- IDs (of the trip and of test participant) should be pseudonymized.
- The personal data about the driver, passengers or test participants are not shared to the CDB.
- The behavior of the single ADF implementations should not be detectable/rebuild. This was achieved by the fact that vehicular sensor data are not uploaded to the CDB as time series but as summarized performance indicators, which are described later.

4. Objective Data Processing Tool Chain

This chapter describes the architecture and the design of the data processing toolchain implemented by L3Pilot, showing examples of algorithms designed to implement this workflow. The chapter focuses on the modules to which development the candidate contributed. A more complete description of the toolchain can be found in [56] and [68]. The CDB framework will be described independently in Chapter 5.

The data toolchain was developed by SP5 (“Pilot tools and data”), particularly these partners: Institut für Kraftfahrzeuge (IKA) RWTH Aachen University, Jaguar Land Rover, Chalmers University of Technology (SAFER), VTT Technical Research Centre of Finland, TNO - Netherlands Organization for Applied Scientific Research, Ford, FEV Gmbh, Institute of Communication and Computer Systems (ICCS), University of Genova.

The specific contribution of the candidate to the modules presented in this chapter is summarized in Table 3.

Table 3 Contribution of the candidate to the developed module within SP5

Module	Role of the Candidate
Common Data Format	Only contributed to the discussions within SP5
Data Conversion Tool	Independently tried to code and test a conversion tool on MATLAB to have more experience and efficient discussion on how to convert and store data into HDF5.
Data Post-processing and Enrichment - DMs and Driving Scenarios	Contributed to the computing of some DMs and driving scenarios, particularly the ones presented in the subsection 4.3

4.1 Overall Data Management Workflow

L3Pilot defined a multi-layer data processing workflow, beginning with raw data acquisition from the vehicles and ending with the analysis of the shared data [62,63].

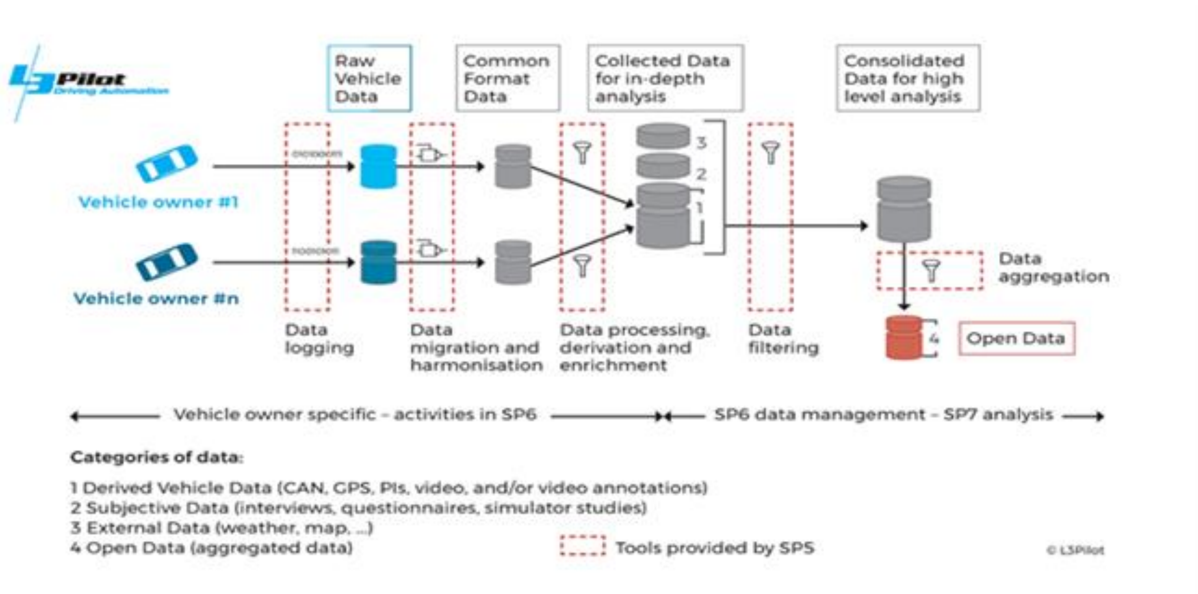

Figure 4-1 High level schema of the overall L3Pilot data management architecture [62,63]

Figure 4-1 highlights four layers characterizing the L3Pilot data processing workflow. The left-most one is the vehicle owner layer, involving proprietary data logged from the vehicle. Filtered data, according to the vehicle owner confidentiality requirements and policies, are then converted to the CDF, to tackle the variety of proprietary data sources. Afterwards, data are post-processed to enrich vehicular signal time series of a single trip with computed derived measures (DMs) and detected driving scenarios. These measures are fundamental for the computation of the CDB PIs (See chapter 5). Selected partners (one selected partner for each vehicle owner) support data owners to execute the scripts for extracting indicators and events and to verify their data ‘quality running the video annotation, data quality checking tools. Nonetheless, such partners are also consulted on test procedures, analysis of signal quality and verification of driving.

4.2 Data Conversion to a Common Format and Storage in HDF5

A key point in the project concerned the processing and the analysis of data from a variety of heterogeneous sources and with a variety of tools. Thus, a fundamental design choice was to define a common format to be agreed and shared by all the project partners. SP5 (“Pilot tools and data”), particularly JLR, IKA-RWTH Aachen Univ., Chalmers Univ., VTT, defined the L3Pilot Common Data Format (CDF), which makes it possible to use a comprehensive toolchain for all analysis steps, following the transfer of data from the vehicle owner up to the delivery to analysts.

From a methodological viewpoint, the format emerged as a combination of a bottom-up approach, stemming from the need to include all the signals necessary to answer the project’s RQs, and of a top-down approach, due to the need for generalizing and abstracting the data structure for future projects.

The first step in the data management chain consists of logging raw data, in a proprietary format, from the vehicular communication buses. Logged data are then converted in the L3Pilot CDF. Conversion is done through MATLAB, Python or C++ scripts. This conversion module, developed by IKA RWTH Aachen University, produces one file for every test trip. The file contains all the information related to that trip, apart from the videos recorded by the several cameras installed in the vehicle (front camera, driver face and upper body, driver hands, driver feet).

L3Pilot SP5 (“Pilot tools and data”) decided to store the CDF data in HDF5 [64], a binary file format characterized by its ability to contain and compress large and complex structured data [65].

HDF5 also includes a data model and software libraries for storing and managing data. Through the compression, HDF5 enables portability, so facilitating the transfer of large amounts of data. HDF5 supports various programming languages, one of them being Python, which is important for exploiting the rising potential within artificial intelligence (e.g., for automatic scene detection and video data annotation). An .h5 file can be extracted to a set of .csv files and, vice-versa, several tables (or .csv files) can be combined into one .h5 file. The benefit over. mat is in the wider selection of supported tools.

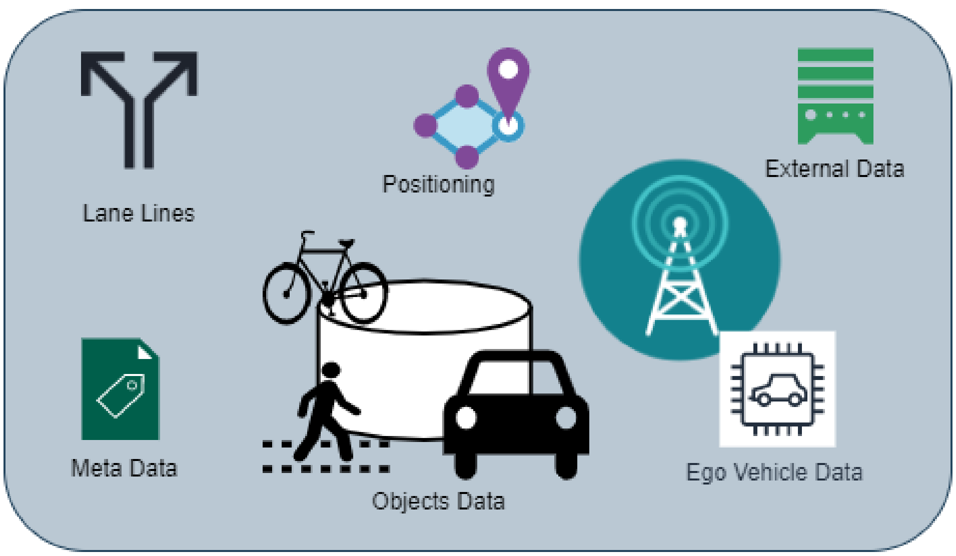

Figure 4-2 The different groups of data inside the HDF5 file

The data inside the HDF5 file are saved in hierarchical data structures. They are grouped in six datasets, illustrated in Figure 4-2. The main signals of the ego vehicle, such as its speed, acceleration and brake pressure are stored in the "egoVehicle" dataset. The "laneLines" dataset saves information about the lane markings, e.g., the distance to the lane markings and their type and the curvature. Dynamic objects surrounding the ego vehicle and their properties, such as speed and distance, are saved in the "objects" dataset. The objects can be trucks, pedestrians, bicycles or vehicles and the “objects” dataset contains information on them like their lateral and longitudinal position with relative to the ego vehicle, their classification etc.

The “positioning” dataset contains information (from a global navigation satellite system (GNSS), e.g., GPS) about the position and heading of the vehicle and the number of satellites. The “metadata” dataset contains context information about the trip, such as the driver type, passengers, vehicle length and width, timing, and experimental condition (test or baseline, fuel type).

As it is important to capture information about the external environment to know under which conditions the ego vehicle was tested, two important external data sources were identified for L3Pilot: weather and map information. Weather information can be provided by various weather services and is about temperature, precipitation, and cloud coverage. Map data provides information about the number of lanes, speed limits or intersections. These data are saved in the “externalData” dataset and grouped hierarchically under the “map” and “weather” subgroups.

In each dataset, each signal timeline (listed in columns) is recorded with a 10 Hz sampling frequency. For each signal, the CDF specifications mandate units, resolution, range of values or enumeration, required frequency, data type and interpolation method (linear or zero-order hold).

L3Pilot has released the CDF as open source [13]. To the best of our knowledge, this is the first time that such a big automotive consortium defined and released as open source a common data format for processing information about automated driving.

4.3 Data Post-Processing and Enrichment

A fundamental processing step, after conversion in the CDF, is represented by data enrichment. This involves the processing, through MATLAB scripts, of the data present in an HDF5 file, to obtain additional information, particularly related to the project RQs. L3Pilot SP5 ("Pilot tools and data") specified three main clusters: Derived measures (DMs), driving scenarios and performance indicators (PIs). The first two are described in the following sub-sections, while PI definition will be presented in section 5.1, about the Consolidated Database, since PIs are the statistical measurements extracted from the automotive signals to be stored in the CDB.

4.3.1 Derived Measures

The derived measures (DMs) are time-series data computed from the collected raw signals. Examples of DMs include the time headway, the time to collision TTC, longitudinal distance between in-front object and ego vehicle, and others. Once computed, the DMs are stored in the "DerivedMeasures" HDF5 dataset, which is added to the original HDF5 file structure.

DMs aim to enrich the dataset by computing compound values useful for the detection of the driving scenarios and the computation of the PIs. DMs were defined and formalized during several workshops among SP3 ("Methodology"), SP5 ("Pilot tools and data") and SP7 ("Evaluation") partners. The main contribution on DMs was by IKA and Jaguar Land Rover. Table 4 shows the DMs implemented in MATLAB by the candidate.

Table 4 Examples of some computed derived measures DMs in the data processing toolchain

Measure	Description	Mathematical Definition
Longitudinal distance to rear vehicle	The longitudinal Distance between the rear object and the rear bumper of the ego vehicle.	$LongDistRV = LongPositionRV - PositionEgoRearBumper$ (1)
Take Over Time	The time interval between takeover request (TOR) and first driver-initiated intervention. The intervention, is defined as the first conscious input, either braking or steering [66].	$TakeOverTime = \mathrm{time}(ManualIntervention) - \mathrm{time}(TOR)$ (2) $ManualIntervention(i) = \begin{cases} HandsOnDetection > 0 \mid\mid \\ BrakePedalPos > 0 \mid\mid \\ ThrottlePedalPOs > 0 \end{cases}$ (3)
Longitudinal Distance Lead Object	The longitudinal distance between lead (in-front) object and the ego vehicle (Figure 4-3).	$LongDist_{leadobject} = LongPos_{leadobject} - FrontBumperPos$ (4)
Longitudinal Distance Rear Object	The longitudinal distance between the rear object and the ego vehicle (Figure 4-3)	$LongDist_{rearobject} = abs(LongPos_{leadobject} - RearBumperPos)$ (5)

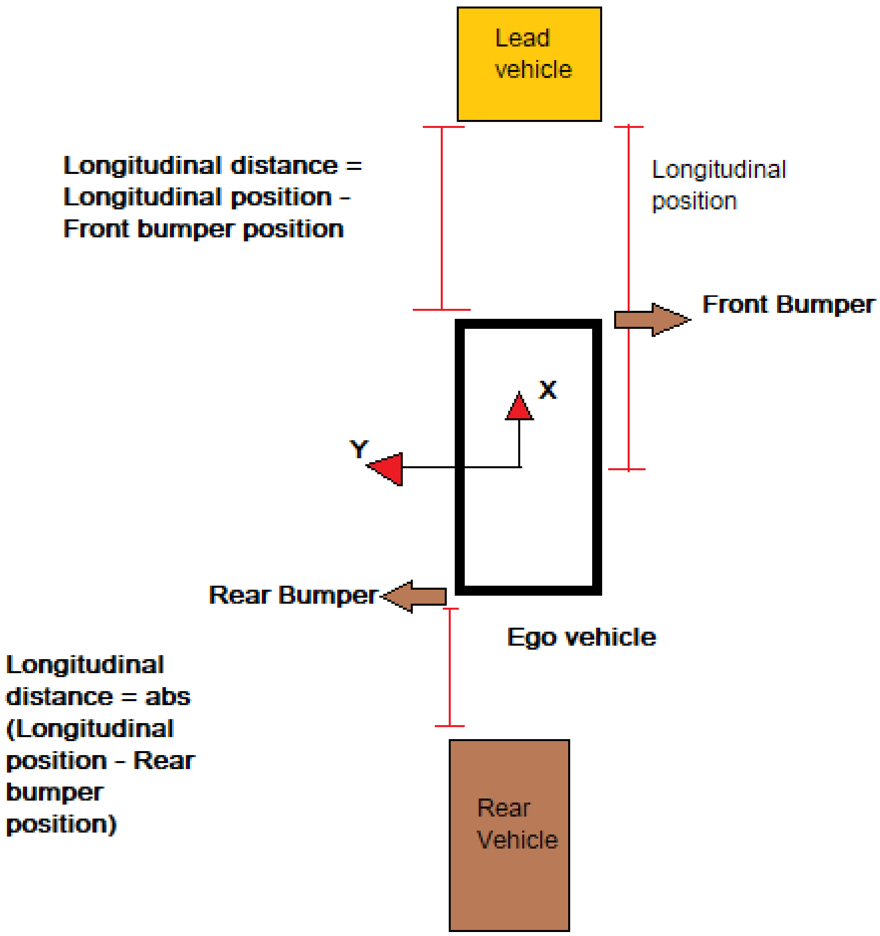

Figure 4-3 The longitudinal distance of the front / rear object to the Ego Vehicle DM

4.3.2 Driving Scenarios

A high number of traffic scenarios may occur during a drive on public roads. Their recognition (achieved by analyzing the raw signal timelines) is necessary, since the L3Pilot research question differentiate the analysis based on the different driving scenarios. Examples of scenarios recognized by L3Pilot include *uninfluenced driving*, *lane change*, *merge*, *cut-in*, *approaching a leading vehicle*, *approaching a static object*, *following a lead vehicle*. Thus, the data toolchain implemented by SP5 (“Pilot tools and data”) includes a module which detects the driving scenarios and saves them in the scenario dataset of the enriched HDF5 file. Scenarios were defined and formalized during several workshops among SP3 (“Methodology”), SP5 (“Pilot tools and data”) and SP7 (“Evaluation”) partners. The main contribution on driving scenario development was by IKA and Jaguar Land Rover. The following sub-sections provide details on the scenarios implemented in MATLAB by the candidate.

4.3.2.1 Lane change of a lead vehicle

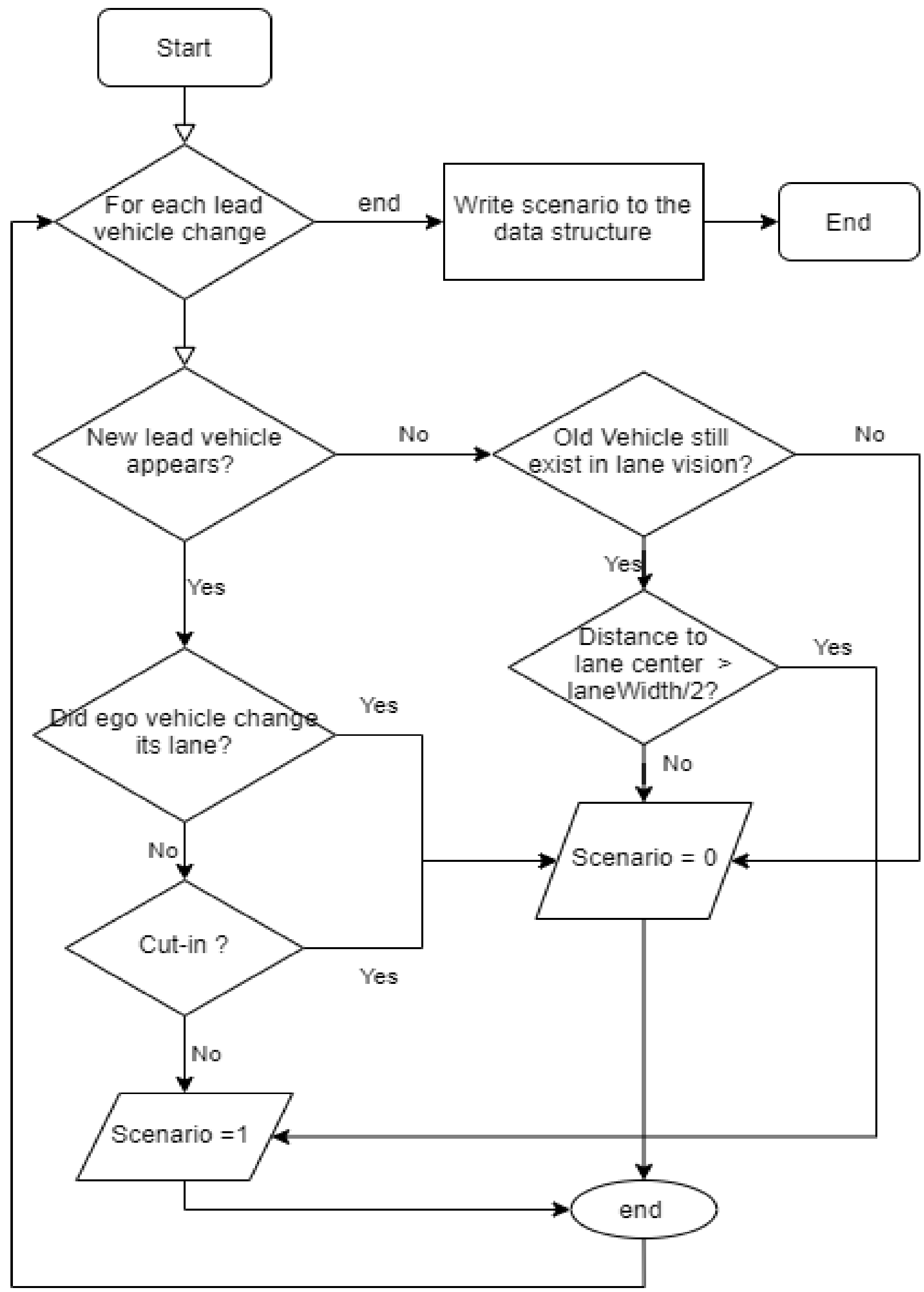

Figure 4-4 Flow chart of the lead vehicle' lane change scenario

Figure 4-4 presents the flow chart of the algorithm we designed to detect the scenario of the lane change of the lead vehicle. It was challenging to detect this scenario as the in-front object varies along the trip and sometimes can be a static object. Besides, it can be a pedestrian or a vulnerable road user with regards to the data object's classification value available in the object dataset timeline. The algorithm thus loops over each time the lead vehicle object changes its identifier (lead vehicle ID), then translates this change into possible scenarios: either a new lead vehicle appears, or the current lead vehicle had disappeared from the lane vision of the go vehicle. Thereby, the algorithm follows the below steps:

1. First, the algorithm loops over all the time instances the lead vehicle object identifier changes:

$$\mathrm{d/dt}(LeadVehicleID) \;\cong\; 0 \tag{6}$$

2. Then, the reason behind the ID change is investigated. There could be many reasons, but the scenario should trigger only when the lead vehicle changes its lane.

   a) If a new lead vehicle appears, the algorithm verifies that the ego vehicle did not change the lane during this time scope, neither the new lead vehicle cut in the lane of the ego vehicle. The latter can be verified by checking the longitudinal distance between the ego and the lead vehicle is greater than a defined threshold:

$$(LongDistance(t_i) > DisThreshold) \mid LongDistance\ (t_i) > LongDistance(t_{i-1}) \tag{7}$$

b) Otherwise, if no newer vehicle is in front, the algorithm checks whether the previous lead vehicle still appears in the lane vision of the ego vehicle. Then, it verifies its distance to the lane center is greater than the half of the lane width. Therefore, the scenario triggers.

$$DistanceToEgoLane(t_i) > LaneWidth(t_i)/2 \tag{8}$$

3. Finally, the results are stored in an array of time instances, which is saved back to the dataset.

The function signature is then defined, as shown in Table 5 (input parameters) and Table 6 (output parameters).

Table 5 Input parameters for the Lane change of lead vehicle's algorithm

Input Parameters	Description
Data	The struct containing all the datasets of the L3Pilot CDF, i.e., ego vehicle, objects, lanes, and positioning. This parameter is common for all the function.
DistThreshold	The Distance Threshold accepted for Lane change scenarios. The distance to the ego vehicle helps to discern the current vehicle in front did not cut in the lane of the ego vehicle. Thus, we consider the lead vehicle made a lane change only if the longitudinal distance is greater than the threshold.

Table 6 Output parameters for the lane change of Lead vehicle's algorithm

Output Parameters	Description
Data_out	A struct containing the input data and, in addition, the new scenario in the corresponding scenario struct, i.e., Data_out.scenarios.LaneChangeLeadObject

4.3.2.2 Following a Lead Vehicle

This function detects the *following a lead vehicle* scenario.

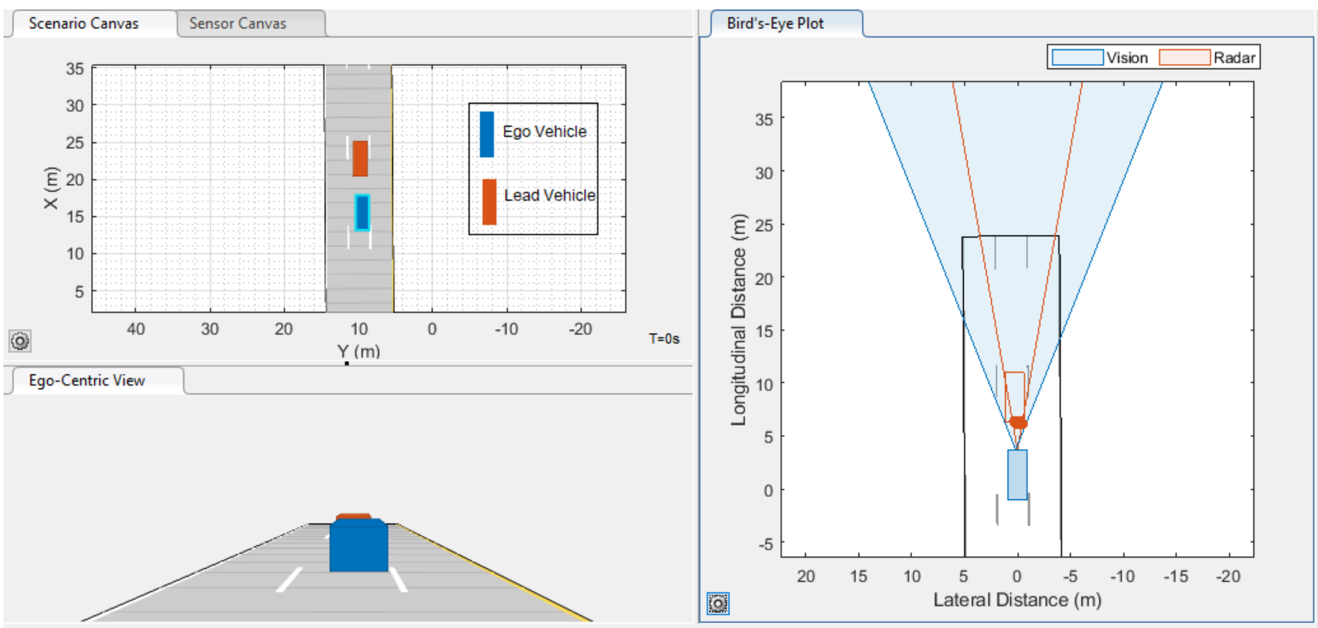

Figure 4-5 Graphical representation of the following a Lead vehicle scenario.

Figure 4-5 illustrates three presentations of the ego vehicle following another lead vehicle' scenario. The Ego-centric presents the chase view by the in-front camera. The Brid's-Eye plots 2-

D coverage areas around the ego vehicle by the mounted sensors; the in-front camera and radar detect the in front objects, measures their velocities and the distances to them. The algorithm detects the potential candidates for following a lead vehicle. It tracks only if:

1. A lead object is detected by the sensors mounted on the ego vehicle.
2. The lead vehicle is driving on an accepted tolerance speed above or below the speed of the ego vehicle.
3. The longitudinal distance between the two vehicles is very short.

The mathematical formulation of the problem was defined as it follows:

$$LeadVehicleID > 0 \,\&$$
$$\| LeadVehicleVelocity \| > EgoSpeed \pm Tolerance \,\&$$
$$LeadVehicleDistance < Threshold(\mathrm{dx}, \mathrm{dv}, \mathrm{v}) \tag{9}$$

The function signature is then defined, as shown in Table 7 (input parameters) and Table 8 (output parameters).

Table 7 Input parameters for the following a lead vehicle function.

Input Parameters	Description
Data	The struct containing all the datasets of the L3Pilot CDF, i.e., ego vehicle, objects, lanes, and positioning. This parameter is common for all the function.
SpeedTolerance	The tolerance above or below the ego speed that is accepted for vehicle following scenarios.
THW	The time headway for the distance to the lead vehicle for following a lead vehicle scenario

Table 8 Output parameters for following a lead vehicle function.

Output Parameters	Description
Data_out	A struct containing the input data and, in addition, the new scenario in the corresponding scenario struct, i.e., Data_out.scenarios.FollowingALeadVehicle

Then, the function is expressed in pseudo-code and finally implemented in MATLAB. The steps of the following a lead vehicle scenario detection are the following:

1. Set the default input values if the relevant parameters were not passed.
2. Check for necessary fields.
3. Loop over all the lead vehicles raw data and, for all the time samples in which a new lead vehicle is detected, check whether the difference between the lead and ego vehicle velocity is within a tolerable range, as defined in (9).
4. Elicit the tolerable distance headway from the time headway by applying this formula:

$$DistHeadway = \mathrm{THW} * Data.egoVehicle.VehicleSpeed(idx) \tag{10}$$

5. Check that the longitudinal distance between the two vehicles is not exceeding the tolerable distance headway for a *following a lead vehicle* scenario.
6. Write the *following a lead vehicle* scenario to the scenario dataset and finally pass the updated overall dataset as the output parameter.

4.3.2.3 Cut-in

Figure 4-6 and Figure 4-7 illustrate the *Cut-in* scenario.

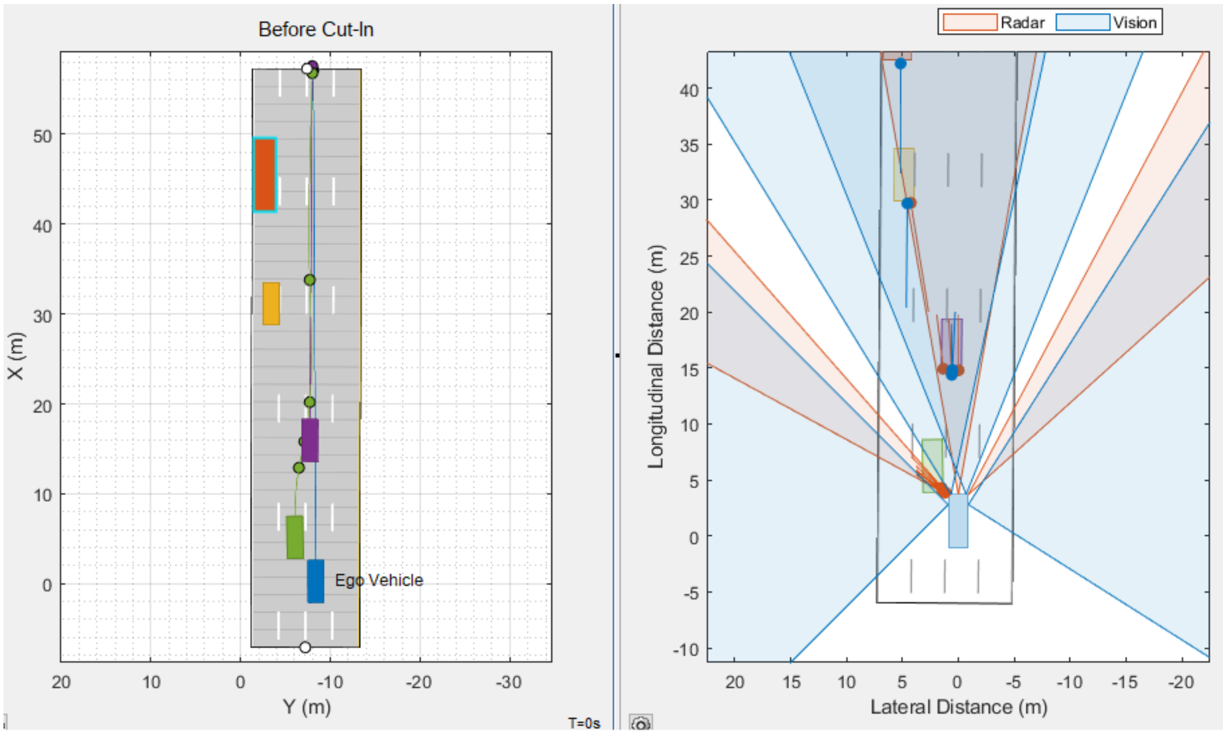

Figure 4-6 Graphical representation of the vehicle (green) cutting-in the trajectory between the ego vehicle (blue) and the lead vehicle (violet)

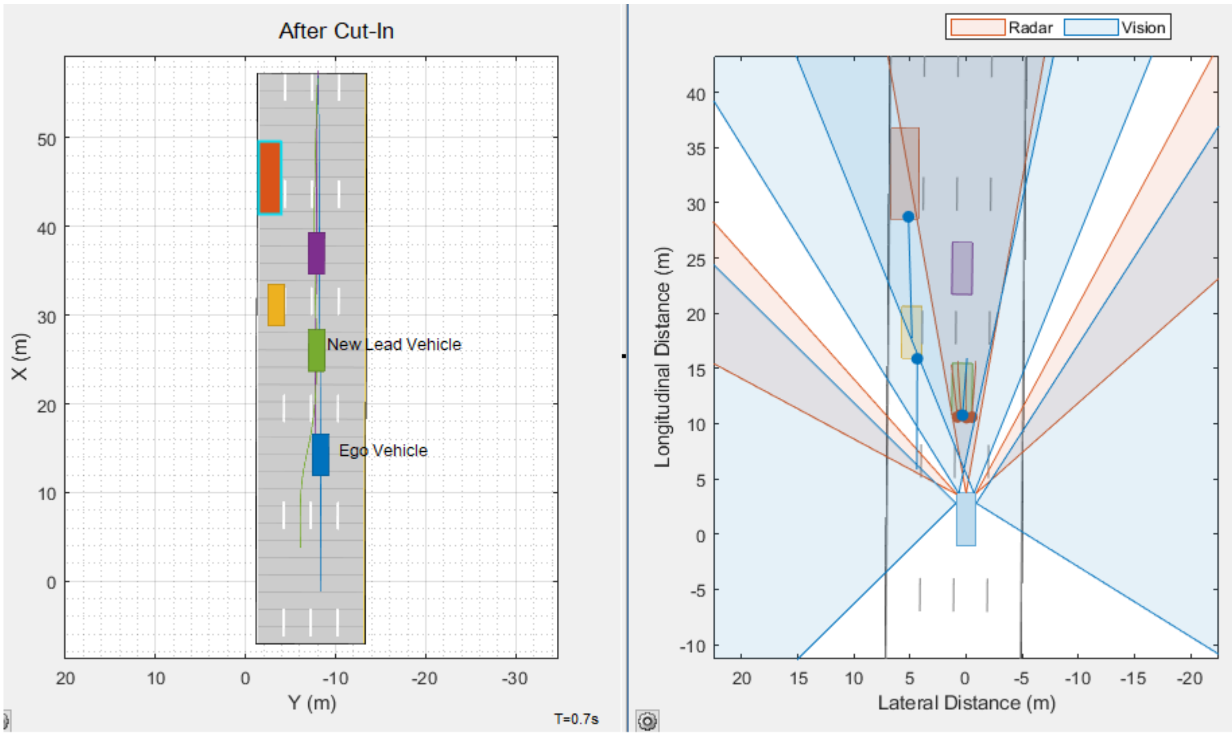

Figure 4-7 Graphical representation after performing the cut-in scenario by the green vehicle.

First, the goal of the function is defined, which, in this case, is to detect a situation in which a vehicle is changing its lane to the lane of the ego vehicle, such that the new resulting scenario for the ego vehicle will be either Following a lead vehicle or Approaching a lead vehicle. The mathematical formulation of the problem was defined as it follows:

$$\mathrm{d/dt}(LeadVehicleID) \cong 0 \,\&$$

$$NewLeadVehicleDistance < ExclusionDistance \,\&$$

$$NewLeadVehicleVelocity < SpeedThreshold \,\&$$

$$NewLeadVehicleOrigin == \{LeftLane, RightLane\}$$

(11)

The algorithm considers the cases where a new lead vehicle is detected. For a cut-in scenario to be detected, some prerequisites are mandatory. First, the distance of such vehicle from the vehicle under test must be less than the exclusion distance (a predefined threshold). Second, its absolute speed should be less than a speed threshold. Finally, the vehicle must be changing its lane coming from either the left or right lane, with respect to the lane of the ego vehicle.

The function signature is then defined, as shown in Table 9 (input parameters) and Table 10(output parameters).

Table 9 Input parameters for the Cut-in function.

Input Parameters	Description
Data	The struct containing all the datasets of the L3Pilot CDF, i.e., ego vehicle, objects, lanes, and positioning. This parameter is common for all the function.
SpeedThreshold	The Speed Threshold accepted for Cut-in scenarios. Vehicles with a speed higher than this threshold are not considered as making a cut-in
ExclusionDistance	The exclusion distance starts from the ego vehicle to the end of the exclusion zone. Vehicles that are farther from the ego vehicle than the exclusion distance are not considered as making a cut-in
DeadPeriod	The period that will be excluded from the calculation
WindowSize	The length of the period where lateral positions of the newLeadVehicle are checked for detecting a cut-in
LateralDistance Threshold	The threshold for the lateral distance

Table 10 Output parameters for the Cut-in function

Output Parameters	Description
Data_out	A struct containing the input data and, in addition, the new scenario in the corresponding scenario struct, i.e., Data.out.scenarios.CutIn

Then, the function is expressed in pseudo-code and finally implemented in MATLAB. The steps of the Cut-in scenario detection are the following:

1. Set the default input values if the relevant parameters were not passed.
2. Check for necessary fields.
3. Preparatory calculations. This involves the creation of the Cut-in scenario array (one entry for each sample in the timeline), which is initially zeroed. Then in a later step the newLeadVehicle_Indices vector is defined, which marks all the time samples where a new lead vehicle is detected.
4. Iterate over the time samples in which a new lead vehicle is detected:
    a) Get the index of the object matching the lead vehicle.
    b) Check that the object is not empty and that its longitudinal distance and absolute velocity are below the given thresholds.
    c) Get the mean of the lateral position of this object starting from (current time - WindowSize) until (current time - DeadPeriod). If this mean is greater than LateralDistanceThreshold, Cut-in is considered from the left. If this mean is lower than -LateralDistanceThreshold, Cut-in is considered from the right. Otherwise, no Cut-in is detected.
5. Write the Cut-in scenario to the scenario dataset and finally pass the updated overall dataset as the output parameter of the function.

If, in 4b above, the object is empty, the execution is interrupted as this is not a *Cut-in* scenario instance. However, if the id of the former lead vehicle still exists, then it is a lane change of the lead vehicle; otherwise, the lead vehicle left the view range.

5. Consolidated Data Base (CDB)

The final step of the data processing involves the preparation of information for the CDB, which was developed by partners of SP6 ("Piloting") and SP5 ("Pilot tools and data"). The goal is to collect aggregated and pseudonymized information from all the HDF5 files and make them available to the whole consortium (while the "selected partner" analysis is restricted to each specific vehicle owner) to support high-level impact analysis.

From an architectural point of view, a platform for supporting project-level data storage and retrieval was developed by the University of Genova. The platform relies on a MongoDB non-relational database, which is accessed through a Node.js application programming interface (API) (Figure 5-1). The platform, which is based on the open source Atmosphere framework [67], exposes a set of RESTful APIs [68,69] for inserting and retrieving data [70]. A web Graphical User Interface (GUI) has been implemented by the University of Warwick to allow a user-friendly access to data. Different user roles have been defined for administrators, vehicle owners, and analysts. Such roles implement different data read/write rights, to meet the project information confidentiality requirements. An Uploader tool has also been created, to support efficient checking (e.g., for duplicates) and batch upload and download of data.

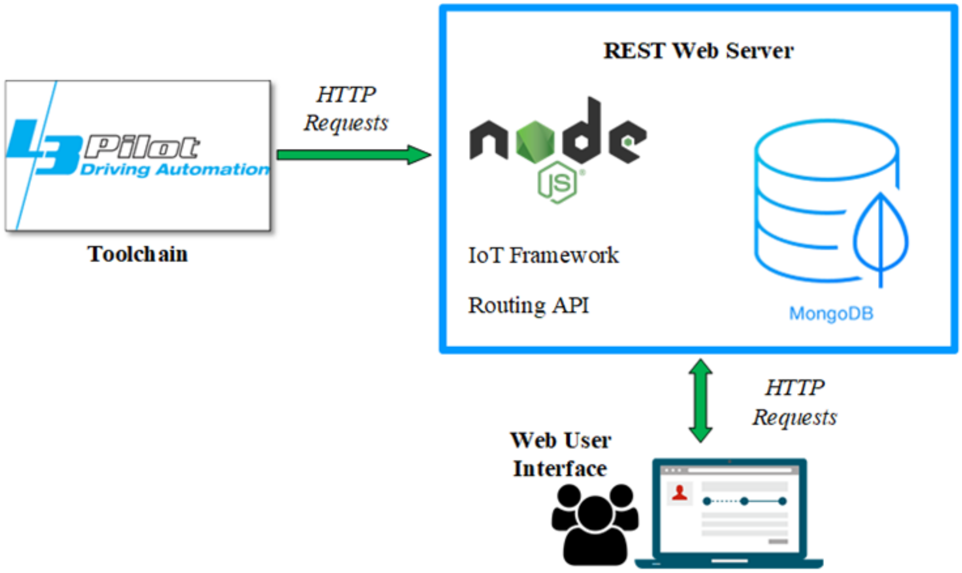

Figure 5-1 The CDB overall architecture [14]

As the personal contribution of the author in this scope of work focused on data processing and aggregation for the CDB, this chapter will present the CDB-aggregator module, which is the responsible for computing the PIs for vehicular data, and the subjective data module. PIs were defined by L3Pilot SP3 "Methodology", SP5 "Pilot tools and data", SP7 "Evaluation", as described in section 3.3. Details on the CDB application programming interface (API) back end, the uploader tool, the user roles and access and the GUI are described in [15].Table 11 outlines the main modules of the CDB presented in this chapter and highlights the contribution of the candidate.

Table 11 Contribution of the candidate to the CDB modules

Module	Role of the Candidate
CDB-Aggregator Objective PIs	Contributed to the computation of the PIs for vehicular data
Questionnaire Management Tool	Implemented the questionnaires in LimeSurvey and contributed to the methodological refinements with the University of Leeds.
Mapping to CDF	Proposed the creation of mapping files that were necessary to link the CDB-aggregator to the CDF, and contributed to their development
Subjective Data Quality Check	Computed the algorithm to check the quality of the subjective data and contributed to the methodological requirements with Univ. of Leeds
CDB-Aggregator Subjective PIs	Developed the module

5.1 CDB Objective PI Computation

To meet the methodological goals for the CDB objective (i.e., vehicular) data, the CDB PI computation step consists in synthesizing the vehicular time series so that the CDB stores only high-level information that allows tackling the project RQs, without compromising the confidentiality of the single-vehicle owner companies.

This stage is undertaken by the CDB-aggregator module, which processes HDF5 files (one per each trip), containing the original time series formatted in CDF and enriched through the computation of the Derived measures and Driving scenarios (Chapter 4). The output of the CDB-aggregator module is represented by a set of .json files storing the computed PIs. Processing an input HDF5 file, the Aggregator produces one .json file for each one of the four PI types defined in Table 2 (i.e., Trip PI, Scenario Instance PI, etc.). The .json files are ready to be uploaded to the CDB, for instance through a well-established Application Programming Interface (API) client such as Postman, or, better, through the Uploader [66]. The same information contained in the .json files is also saved in corresponding .csv files that are more easily readable by the analysts.

The CDB-aggregator module consists of a set of MATLAB scripts. Figure 5-2 provides a high-level outlook of the programme, with three main phases: initialization, reading signals from the input HDF5 file; processing loop; and a final saving of the four types of PIs. The processing loop is the core of the programme, as it processes the time series and segments the computation of the PIs according to the context information presented in Subsection 3.3. First, the experimental condition is considered. Then, for each identified segment, the road type is considered. This level of segmentation leads to the computation of Trip PIs. Computation of Scenario Instance PIs and Datapoints requires further segmentation of the timeline based on the detected driving scenarios. Scenario Specific Trip PIs introduce the need for accumulating the indicator values across all the scenario instances in the trip. Similarly, the length of each scenario instance is needed for the Trip PI indicator reporting the percentage of time passed in each scenario within the trip.

An example of the resulting segmentation is reported in Figure 5-3, where we can see that eight different scenario instance PIs have been computed. One-time interval, indicated as Unrecognized 1 (U1), does not produce Scenario Instance PI, nor Datapoints, nor Scenario Specific Trip PI, as a scenario could not be detected there. However, the signal values contained in that segment do contribute to the Trip PI indicators in the ADF on condition.

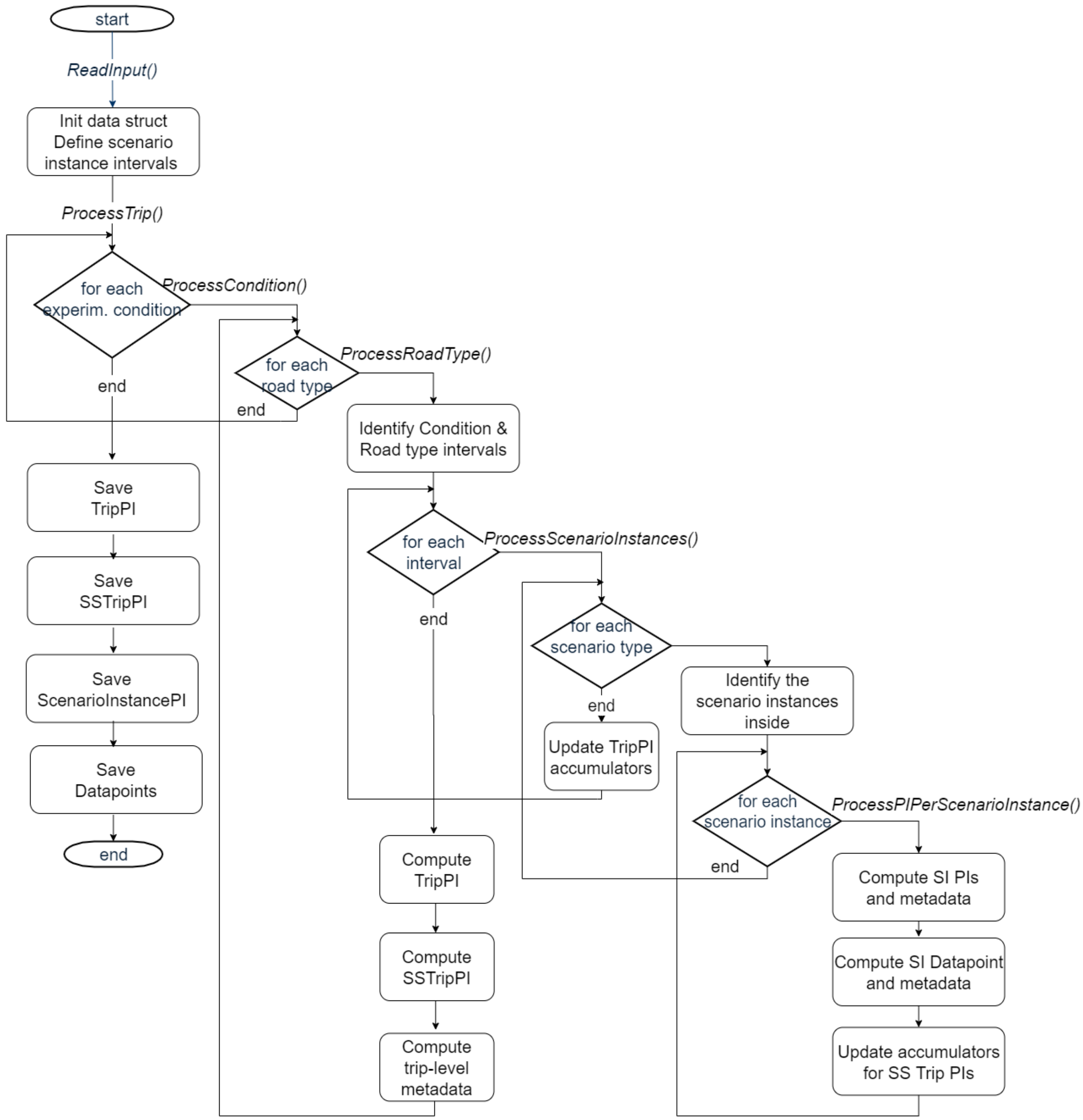

Figure 5-2 High-level flowchart of the PI computation by the CDB-aggregator [68]

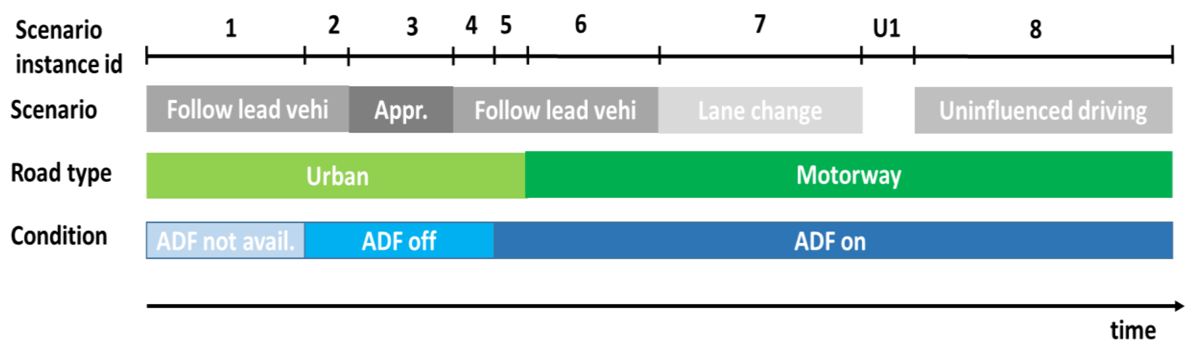

Figure 5-3 Example of scenario segmentation during a trip [68]

5.2 Subjective Data Processing

The second type of data addressed by L3Pilot is subjective data, that were collected through questionnaires defined by SP3 "Methodology" based on the methodological requirements presented in section 3.4. In L3Pilot, test participants were asked to reflect on ADF and report about their test experience, mainly addressing the RQs on user acceptance evaluation and socio-economic impact evaluation. Three questionnaires (motorway, urban and carpark) were designed, corresponding to the three main driving environments, with their relevant ADFs.

Based on the requirements, a reference workflow for processing subjective data was defined to be implemented in each pilot site. Figure 5-4 illustrates the workflow. The first step consists in collecting the questionnaire data. L3Pilot developed a reference implementation (it is described in section 5.2.1) exploiting the LimeSurvey online tool [71], which was selected based on the requirements stated at the end of section 3.4, even if pilot sites could use different tools for questionnaire management. The output of the questionnaire tool is then formatted in the common data format (CDF) for subjective data. This step is described in section 5.2.2 for the reference implementation. Then, a quality check step is performed (section 5.2.3) before the conversion of data to the .json format acceptable to upload to the CDB. Like for objective data, also subjective data are then ready to be accessed by SP7 “Evaluation” analysts.

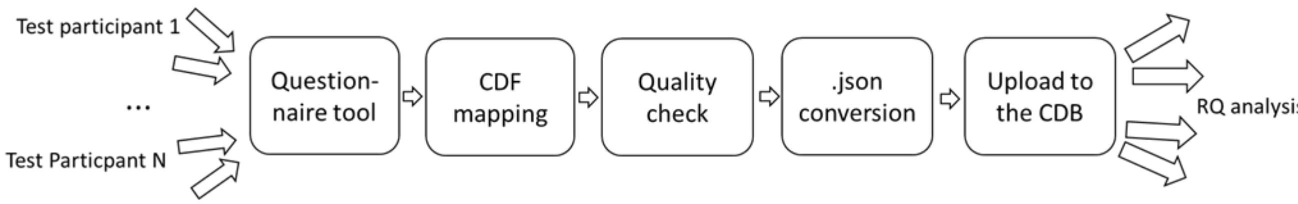

Figure 5-4 Subjective Data workflow in each pilot site

The work of the candidate on subjective data mainly concerned the implementation and refinement of the methodological requirements in collaboration with Univ. of Leeds human factor researchers.

5.2.1 Questionnaire Management Tool (LimeSurvey Implementation)

This section describes the reference questionnaire implementation using LimeSurvey. The implemented questionnaires involved different types of questions, that were handled by LimeSurvey, such as: single choices, arrays, mask questions (e.g., date/time, equation, ranking yes/no, file upload, etc.), test inputs and multiple choices questions.

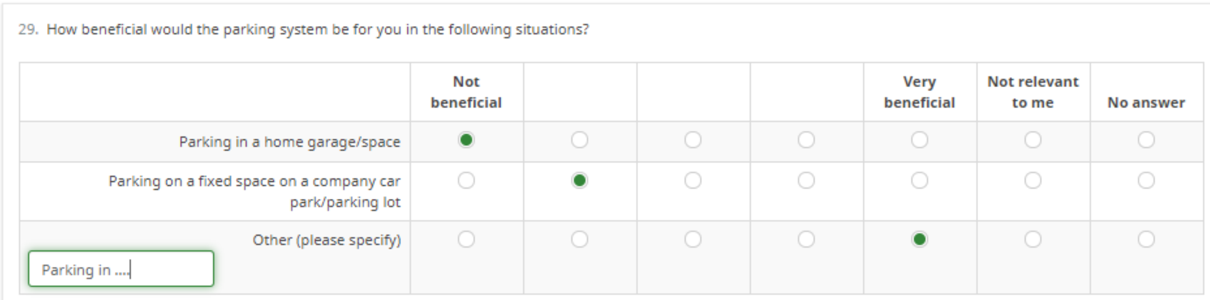

Figure 5-5 Example of customized array' sub-question inputted by the participant.

While LimeSurvey provided valid templates, customization was needed for some questions, for instance, to add text options to numeric arrays or to add text input to sub-questions (typically to allow the participant to specify the "Other" option) as shown in Figure 5-5, or to automatically set values for consistency among answers. Figure 5-6 shows another example of customized arrays where headers are merged on three columns to fit the question' context requirement.

Customizations were implemented by modifying/adding the JavaScript and jQuery source code for the relevant questions. LimeSurvey has a well-established user/developer community that provides extensive support for hand-tailored designs, queries, and extensions.

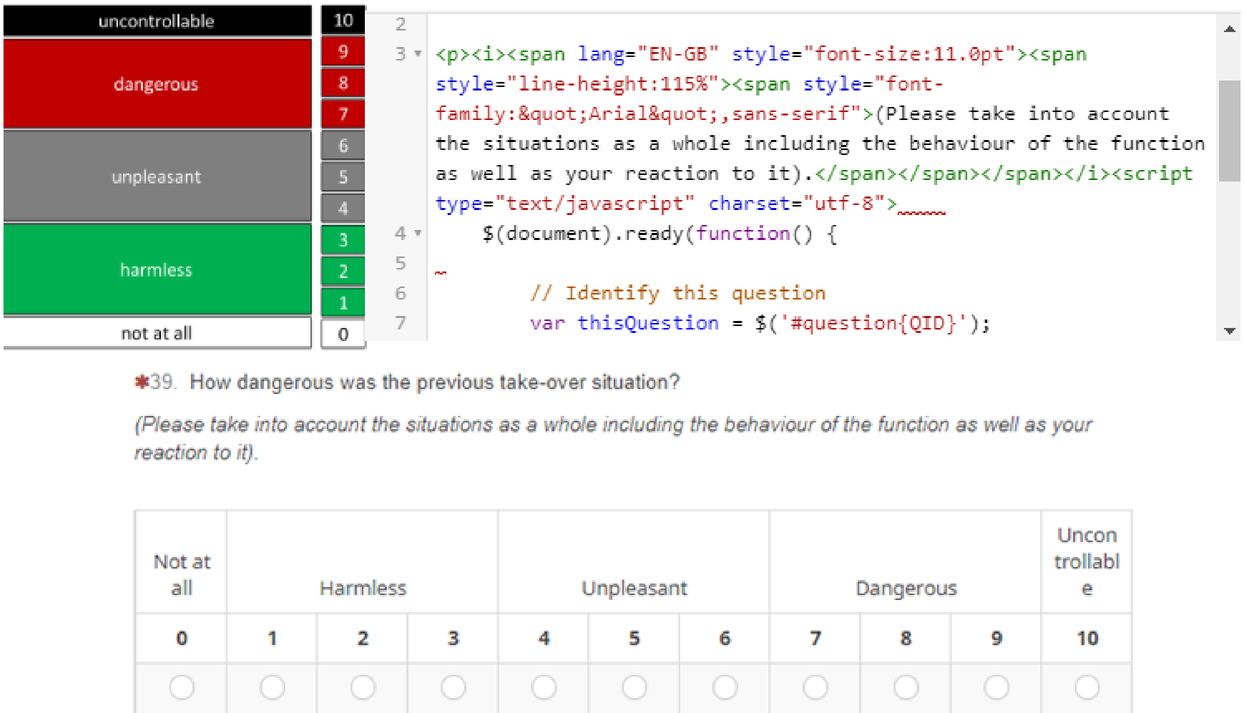

Figure 5-6 Example of customized array. Up left: design requirements of the question. Up right: source code edited behind. Bottom: question's UI

Conditional questions were also implemented, for example to show (or hide) some questions (or their answers), based on the answer to a previous question. Conditions can be added from a user interface or through regular expressions [72]. As an example, Figure 5-7 shows how to set a condition on a question about the household gross income of the participant, where options are grouped in a drop-down list and the context (i.e., the currency) varies regarding his selection of the country of residency.

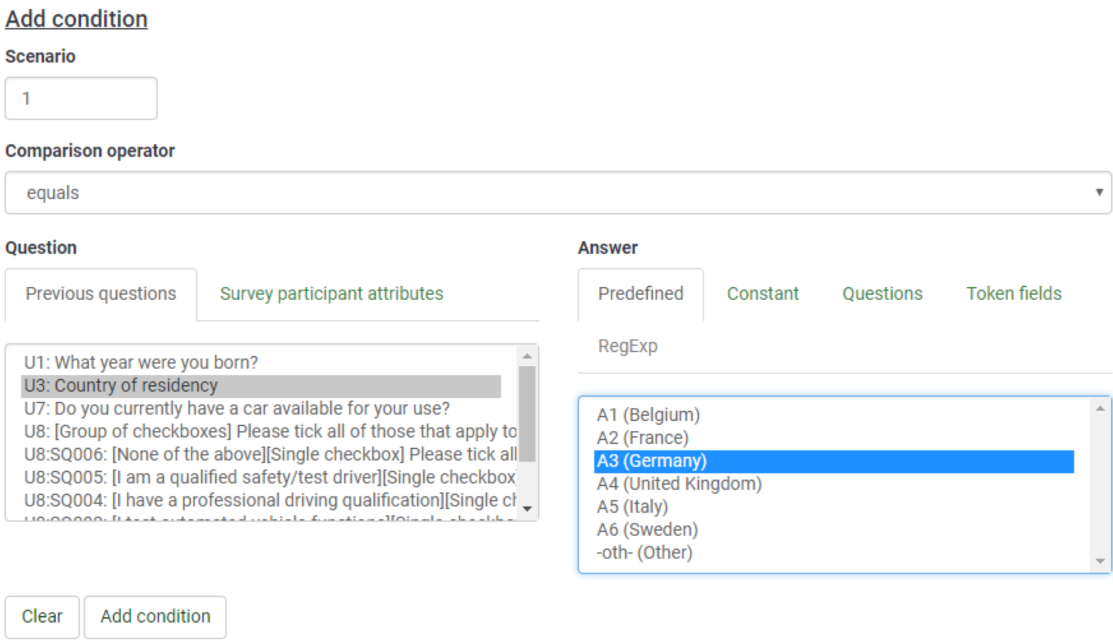

Figure 5-7 Add condition to the question based on previous questions.

Several questions required participant's input validation, to prevent the insertion of incorrect data. For instance, Figure 5-8 and Figure 5-9 show an example of question where participants must provide ordered values and select at least one preference or three at most when ranking transport mode preferences. However, two choices cannot have the same ranking. This is guaranteed by writing validation expressions which are checked at runtime by the LimeSurvey Expression manager. For instance, if the participant selects that he does not take the same trip, the other selections should be automatically dropped as shown in the first row of the same figure.

22. What mode of transport do you typically use for the following trip types? Choose 1-3 often used modes: 1 for the one most used, 2 for the second most used (if applicable), 3 for the third most used (if applicable). Exclude trips made by airplane.

Please make your selections reasonable and valid for each trip type.

	Passenger car	Public transport	Taxi	Motorbike or scooter	Bicycle or walking	I don't take such trips
Commuting	...	...	...	...	...	√
Business travel	1	...	...	...	...	...
Leisure/social	1	1	2	...	...	...
Errands (incl. groceries)	3	...	...	...	...	...

Figure 5-8 Example of questions with input validation

#	Name [ID]	Relevance [Validation] (Default value)
G-5	**Trip Choice** [GID 633]	1
Q-24	**U22** [QID 9863] Array (Numbers) [:]	1 (VALIDATION: ((sum(! (((! is_empty(U22_SQ001_A6) && sum(U22_SQ001_A1, U22_SQ001_A2, U22_SQ001_A3, U22_SQ001_A4, U22_SQ001_A5) == 0) || (is_empty(U22_SQ001_A6) && (U22_SQ001_A1 == 1 || U22_SQ001_A2 == 1 || U22_SQ001_A3 == 1 || U22_SQ001_A4 == 1 || U22_SQ001_A5 == 1 || U22_SQ001_A6 == 1) && ((sum(U22_SQ001_A1, U22_SQ001_A2, U22_SQ001_A3, U22_SQ001_A4, U22_SQ001_A5) == 1) || (sum(U22_SQ001_A1, U22_SQ001_A2, U22_SQ001_A3, U22_SQ001_A4, U22_SQ001_A5) == 3) || (sum(U22_SQ001_A1, U22_SQ001_A2, U22_SQ001_A3, U22_SQ001_A4, U22_SQ001_A5) == 6)) && (is_empty(U22_SQ001_A1) || (U22_SQ001_A1 != U22_SQ001_A2 && U22_SQ001_A1 != U22_SQ001_A3 &&

Figure 5-9 Validation expression in LimeSurvey

Following the completion of questionnaires, test participants' responses results must be exported to the xlsx file format (as in Figure *5-10*).

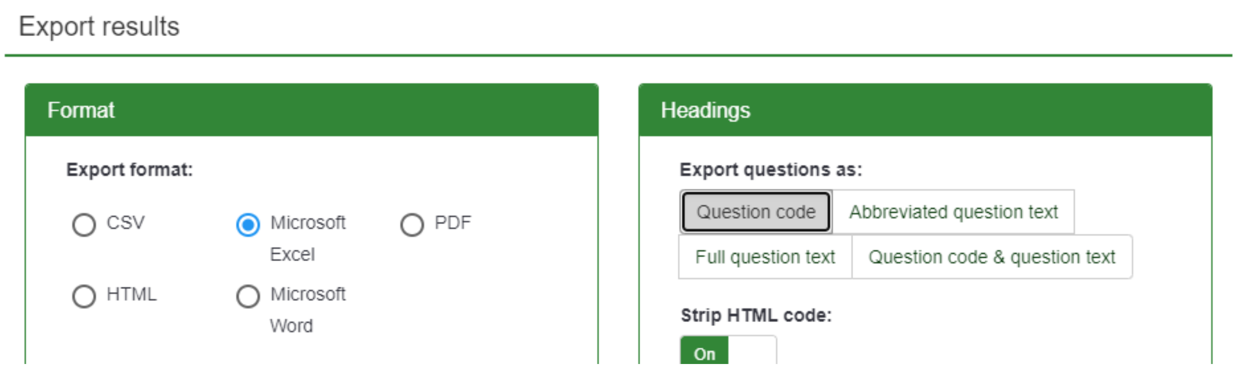

Figure 5-10 *Function to export questionnaire responses to excel format.*

5.2.2 Mapping to CDF

Although selected partners/pilot sites responsible could create, edit, or view a survey, SP3 ("Methodology") and SP7 ("Evaluation") mandated that the questionnaire item codes are not changed, as this may allow tracking responses from different pilot sites. To ensure that data could be correctly collected across all the pilot sites, instructions on the questionnaire implementation, administration and metadata were defined at consortium level. This approach ensures that the data output can be integrated seamlessly and transferred to the consortium-wide CDB.

The methodology partners specified all the questionnaire items required to be shared to the CDB. In addition, they set four supplementary items to be added by the pilot sites to entry, such as ADF type, participant ID, participant type, and test type. All such data (questions and possible closed answers), grouped in the three L3Pilot questionnaire types, were encoded in a common data format (CDF) specific for subjective data.

Using the common LimeSurvey implementation had to ensure that all questionnaire items and responses follow the CDF schema. However, some partners used other implementations, and post-editing could introduce errors. Thus, to guarantee integrity of subjective data before uploading them to the CDB, a data format map was prepared for each type of questionnaire, with all the required questions and possible answer codes and ranges, according to the CDF.

Figure 5-11 screens a traffic jam and motorway questionnaire's mapping including the IDs of each question and the code-based text interpretation of possible answers. The output values are used for data parsing of exported answers to their corresponding in the CDF. The merge column specifies conditional questions that need to be merged in one item compatible with the CDF. The upper and lower limits, however, are useful to check that the answer codes are within the acceptable interval limits.

Section	Question ID	Label	output Values	Code Based Text Information Interpretatio	Merge	Lower limit	Upper limited
Supplement Questions	ADF_Type	i. What is the ADF	{1, TJ}...	"1"="TJ" "2"= "Motorway" "3"="Urban" "4"="Parking"		1	4
	Participant_ID_T	ii. What is the parti	{XY000...}...	(It is a random generated 8 digit code)		NA	NA
	Participant_Type	iii. What is the part	{1, Professional (test) driver}...	"1"= "Professional (test) driver" "2"= "Non-professional driver" "3"= "Passenger"		1	3
	Test_Type	iv. What test type	{1, Real Pilot road test}...	"1"= "Real Pilot road test" "2"="Test track" "3"="Driving Simulator" "4"="Wizard-of-Oz"		1	4
stionnaire Questions Items- Pre-piloting questions	TJM1	1. What year were	{A10, 1909}...	1929		1929	2001
	TJM2	2. What is your ge	{A1, Male}...	"1"= "Male" "2"= "Female"		1	4
	TJM4	4. What is the high	{A1, trade/technical/v	"1"= "trade/technical/vocational training" "2" ="university degree"		1	3
	TJM5	5. What is your em	{A1, Employed full-time}...	"1"= "Employed full-time" "2"= "Employed part-time" "3" ="Self-employed"		1	7
	TJM6	6. Could you do pa	{A1, Yes}...	"1"= "Yes" "2"= "No"		1	2
	TJM7	7. Do you have a c	{A1, Yes, (nearly) always}...	"1"= "Yes, (nearly) always" "2"= "Yes, sometimes" "3"= "No or hardly ever"		1	3
	TJM8_A1	[I am an employee c	{0, Not selected}...	"1"= "Y"{selected} "2"= " "{not selected} "0"= " "{not selected"		1	2

Figure 5-11 Mapping for questionnaires on traffic jam and motorway

5.2.3 Data Quality Check

This sub-section describes a MATLAB script developed to capture possible errors in a subjective data input .xlsx file. The process involves three main steps.

First, it verifies that all questionnaire items and responses follow the nomenclature set out in the CDF (See Figure 5-11). Using the common LimeSurvey implementation should prevent this issue. However, there is always the chance that in a local implementation some item names and/or codes deviate from the original. In case item names and responses do not match the CDF specifications, such items are saved in a log file and the user is warned to check and verify the data input.

Second, it checks if the item responses fall within the range set out in the original questionnaire. Let us consider the case of a question with an answer set between 1 and 5 (corresponding to a five-point Likert scale of “Strongly Disagree” to “Strongly Agree”). If a given response falls outside that range, then the user would receive an error message asking for a content check.

Finally, there may be some situations where a pilot site wished to upload a dataset that contains missing values. Data could be missing because there was an error in data collection, or a pilot site has chosen to not collect that particular item. There could also be some errors in the data transfer process between data collection, LimeSurvey, and CDB upload. In order to distinguish the case of data that are known to be missing, pilot sites are asked to fill the corresponding cells with a dummy response (-1). Thus, should a pilot site attempt to upload dataset that inadvertently includes empty cells, the data quality checker would warn him with an error message and save a trace in the error log file for all the missing items.

Further quality checks verifies the data types and the possible number of digits for an acceptable response. Table 12 provides an example of such checks.

Table 12 Data quality check on subjective data based on the CDF.

Item	Data Type	Number of digits	Missing data is acceptable?
Participant id	Numeric and characters	8	No
Year of birth	Numeric	4	-1
TJM14	Numeric and characters, it also supports some special characters like $, € but the system saves only numeric inputs to the Database	30	-1
TJM35	Numeric	3	-1
All other items	Numeric	1 or 2	-1

When an error triggers, the quality checker saves the empty/mistaken answers in the log file.

5.2.4 Subjective Data CDB-Aggregator

The subjective data CDB-aggregator module is responsible for preparing the data in .json structures readable by the CDB. It comprises a set of functions implemented in MATLAB that read the xlsx file (or a set of files in a directory) exported by LimeSurvey and implement the functionalities described in the previous two sub-sections, namely CDF mapping and data quality check.

As anticipated, some pilot sites did not use the LimeSurvey system for collecting data through questionnaires. These sites implemented their own modules to translate their data to the defined

CDF. Once translated to the CDF, also these data are homogeneously processed by the CDB-aggregator.

Figure 5-12 shows the logic of the module, which, for each xlsx input file, loads the relevant CDF map (there are different maps for the three questionnaire types), and verifies that all the questions' items required in CDF are present. Otherwise, an error to the user is issued.

The next step ("Alter Table") performs a translation, when needed, from LimeSurvey alpha-numerical codes into numerical only codes required by SP7 "Evaluation". Then, the algorithm calls the data quality check' function described in section 5.2.3 to verify integrity of the inputted answers. If some errors trigger, the CDB- aggregator stops processing and asks the user to verify the empty/mistaken answers saved in the log file by the data quality checker. Otherwise, the algorithm continues the process and converts the subjective data to json structures uploadable to the CDB.

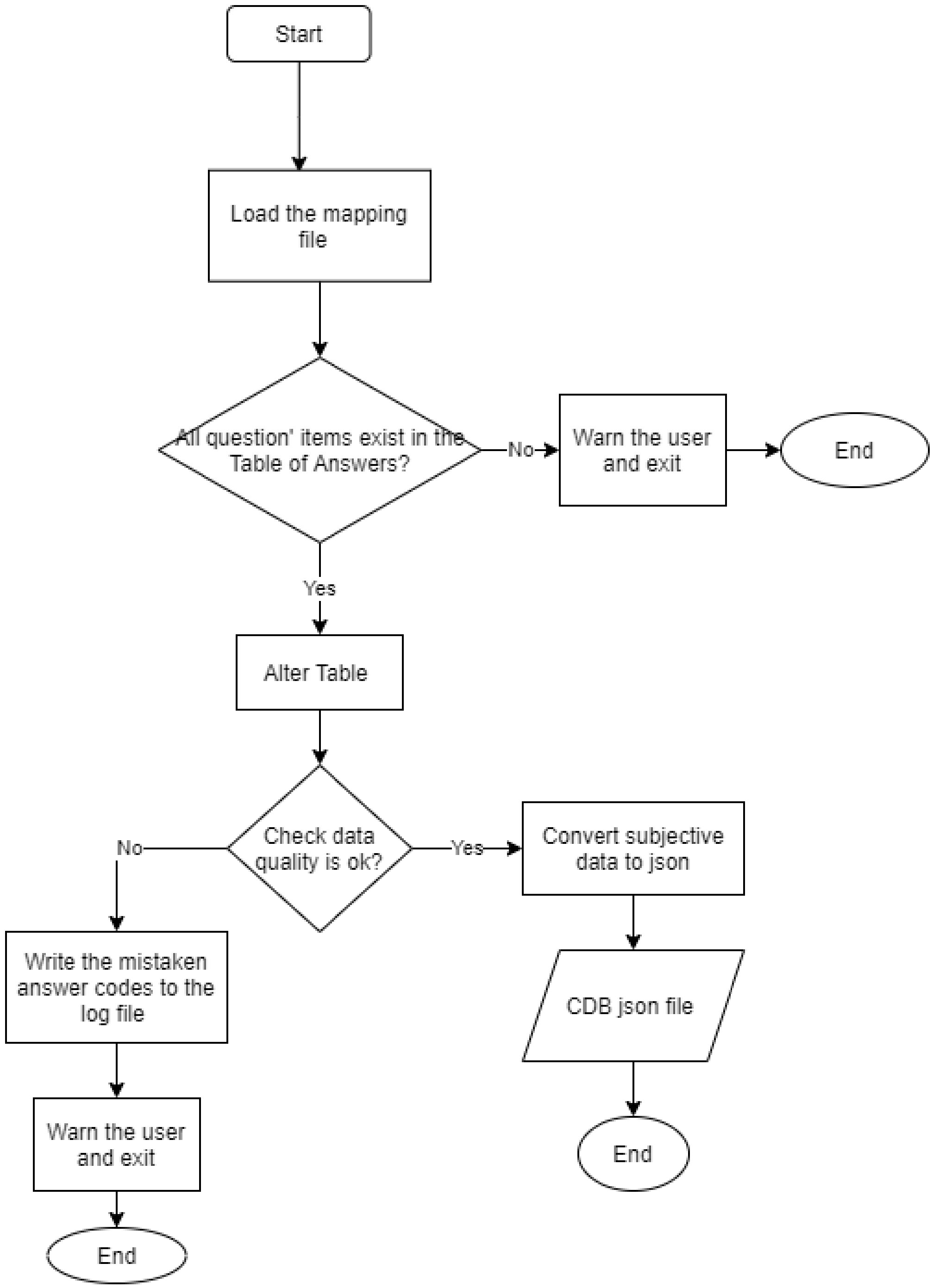

Figure 5-12 Flowchart of the processing of a subjective data file by the CDB-aggregator

6. Deployment and Assessment

As of April 2021, the data toolchain has been developed, tested, and deployed in all the pilot sites within SP6 “Piloting”. Deployment and use of our modules were carefully coordinated between developers and pilot site managers. This careful collaboration, which involved several meetings and provision of detailed deployment and usage instructions in the project’s collaboration tool, was extremely useful to the success of the project. Given a good documentation and presentation of the data tool chain, partners involved in the pilot sites got familiarity with the process, according to their different roles. Various patterns of use could be observed. Vehicle owner companies uploaded and checked their data and analysts accessed and analyzed data from all the pilot sites, but only concerning their specific features (i.e., subjective, or objective data and different types of PIs within subjective data).

The whole data chain was tested in the pre-pilots, that were planned for preliminarily testing the data processing and analysis chain, before full-scale, on-road tests. These tests allowed developers to spot bugs and face challenging situations, that we had not previously considered. Feedback on this from the users helped to improve communication and overall effectiveness, in an iterative process. Not only did the pre-pilot testing of the installations highlight some bugs in the code, but it also enabled the developers to tune the overall process and suggest significant improvements, based on the experience and the analysis of the first sets of (real and synthetic) data. Such suggestions were discussed in the L3Pilot multidisciplinary team and finally implemented, leading to a continuous improvement of the methodology process described in chapter 3 “Methodology and requirements”.

In the following, we go in the details of the deployment of the main modules.

6.1 Subjective Data Modules

The LimeSurvey template file of the questionnaire was delivered to all pilot sites, where the staff had to make the country language translation and/or other customizations according at each pilot site's specific policy. The selected partners/pilot sites were provided with guidelines on how to import the surveys and export the responses.

In terms of the administration of the questionnaire, there have been differences between pilot sites regarding the length and number of drives by each participant. Therefore, the project recommendation was that the questionnaire should be completed after the last test ride, irrespective of whether a driver has multiple drives [44]. Participants to the drive-tests were recruited by the pilot site leaders among ordinary and professional drivers [60].

During pre-pilots it became apparent that some supplementary items (i.e., metadata) should be added to allow a proper assessment by SP7 "Evaluation". These metadata include the ADF type, the participant ID, the participant type, and the test type. Particularly, the pseudonymized ID was necessary to allow pilot sites track their own subjective data after uploading them to the CDB. This ID was employed also as the participant token, which is used by LimeSurvey for the distribution of the questionnaires and to link the answers to the participants. This was implemented by setting up a participant's table in LimeSurvey, linking each test participant personal data (available only to the pilot site staff) with the participant token (to be exported to the CDB), as shown, for instance, in Figure 6-1.

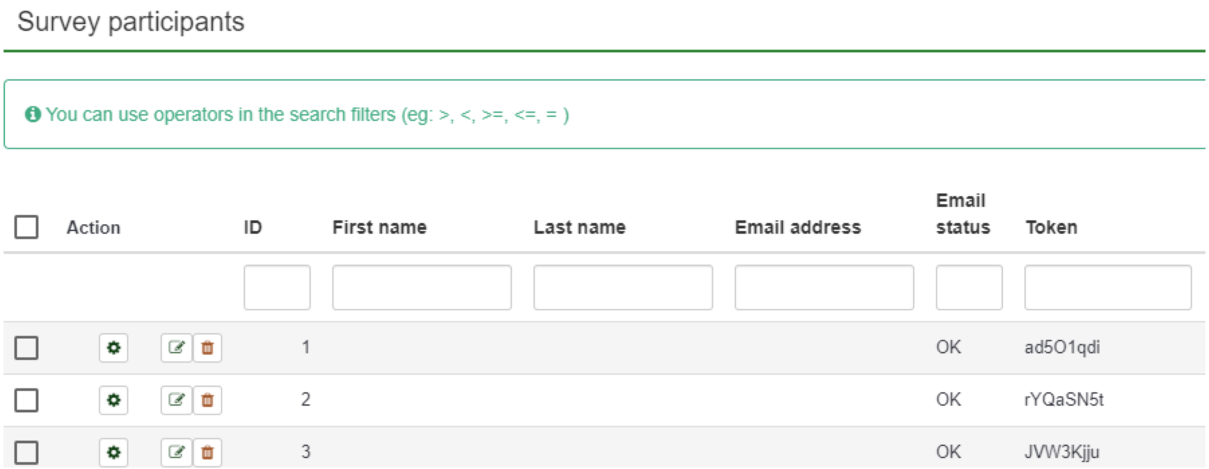

Figure 6-1 Participant's table implemented in LimeSurvey.

To support a correct translation of the questionnaires, it was important to provide pilot site staff with indications on how to make the translations in LimeSurvey without impacting the common format (questions and answers' codes). Instructions were provided to pilot site staff about running the CDB-aggregator module. The staff provided continuous feedback, that was useful to iteratively improve the system, tuning the requirements and their implementation. This stress the importance of designing the system for flexibility and extension.

The partners were generally pleased about the process and response time (in the order of the seconds). The CDF and quality check were found to be very useful. Enhancements could be added to the data management tool chain, through an excellent team collaboration. Table 13 synthetizes feedback from diverse partners about their experience.

Table 13 Feedback on the use of the different modules of the subjective data toolchain

Step	Feedback
The process of implementing the questionnaires in LimeSurvey	The process could be error-prone in the sense that to implement the questionnaires we had to find in Confluence (i.e., the project official collaboration tool) the right version of the questionnaires. Therefore, we implement the latest version available in Confluence, not knowing if this was the right one. The process could have been improved if a direct link to the final version of the questionnaire was provided to everyone. Especially for the post-questionnaire, it was not ready to be immediately implemented because some pilot sites, for some reasons, had to remove some questions. This was not immediate and required some technical support.
Exporting Questionnaires from LimeSurvey	There was a lack of instructions, leading to the fact that some answers were missing because not all the right export options were selected. Handling with this issue required some technical support from our side, which could be avoided if proper instructions had been given beforehand.
Put questionnaire data in the defined common format	It took some effort to convert the data into the common data format because there were some errors in the implementation. E.g., even with our latest version, 8-digit user IDs are only accepted if they contain at least one letter, but it is possible that the hashed ID consists of eight numbers. Such IDs were changed manually before conversion. Some of the manual steps could probably still be improved, like adding of the metadata (these could be parameters given when calling the script) to remove even more possibilities for manual errors. Free text field from LimeSurvey were not read properly from the csv file if the answer had line breaks, so these had to be removed manually. The script could also have capabilities to remove either answer from a single token or answers to questions from all participants.
Converting the data into .json file	No problems were reported. A suggestion stressed that the driving context (i.e., whether it was urban or motorway or parking) should come from the questionnaire data rather than from the filename, as we did for ease of implementation.

In the future, it should be taken care that implemented process can be handled by human factor researchers without deep knowledge in IT, without Admin rights and probably also without access to expensive tools like MATLAB. It is no option that user need e.g., to use their private PC because they are not allowed to install the required tools on their work PC, because of the organization policies.

6.2 Objective Data Modules

All the modules of the objective data toolchain were finally integrated to process the vehicular data. The CDB was deployed in the cloud, together with the web user interface. In parallel, the Uploader was distributed to all the pilot sites. After the login to the CDB, each data row from the input files is tentatively inserted to the CDB provided that the structure of its data matches the corresponding Feature resource, which is the CDB integrity check.

Writing detailed instructions and suggestions in the project's collaboration tools (Wiki or README files in the Gitlab code repository for technical developers, and Confluence pages for all the users) was useful to facilitate the usage of the system.

Important system functionalities have been added thanks to this collaboration. For instance, we initially considered more experimental conditions than those presented in Section 3. There was also a "Treatment" condition, aggregating a trip's measurements independent of the status of the ADF—it is sufficient that the ADF is on the vehicle, as opposed to the baseline condition. However, analysts asked to remove this condition, to reduce the amount of data to be processed.

A conceptual problem found at the beginning of the deployment was that, if the experimental condition changes within an occurring scenario, two scenario instances are created and uploaded to the database although there is only one scenario occurring. This is fine in some scenarios. For others, however, this gives wrong results for, e.g., the duration of a *lane change* or the standard deviation of the speed during *following a lead vehicle*. The problem became apparent when looking at *lane changes*. During the recordings, *lane changes* are often not performed by the ADF but by the safety driver, or at least signs off on them. In the aggregated data this leads to three scenarios that are uploaded to the database and that are evaluated at the end, which is not the desired output. L3Pilot analysts thus introduced the concept of “partial” (uninfluenced driving, following a lead vehicle) and “complete” (all the others) scenarios. For partial scenarios, splitting them up, due to an intervening condition change (e.g., from ADF on to ADF off), is fine. For all others, the complete scenario instance is always needed, no matter the condition changes during the scenario. Moreover, all the transitions of conditions that may occur during a complete scenario need to be traced. These additions were handled by adapting the CDB-aggregator workflow reported in Figure 5-2.

6.3 Achievements and Discussions

Finally, the whole data toolchain has been employed in the L3Pilot tests sites, from Italy to Sweden, both in its cloud and local versions. Vehicular sensor data are now being processed by six impact analysis teams and as many traffic analyses teams, while subjective data by three teams, to respond to the research questions. In the following we go in the detail of some major achievements.

A test showed that the L3Pilot CDF stored in HDF5 files is more efficient than formats well established for FOTs, such as csv (82% size reduction). On the other hand, it performs almost as well in terms of memory efficiency as the MATLAB proprietary format (9% size increase), while being independent of the software used [64]. Portability has already been successfully experienced using various tools that process the CDF files in different environments: Windows or Linux, and using Python, R, Java, or MATLAB. The binary format requires specific tools for accessing/adding data in HDF5 files, but this is considered a minor limitation. L3Pilot contributed new open-source code to the HDF Group by improving the Java support of the format (e.g., handling complex variables, which entailed a table within a table).

Using a common format among different vehicle owners was deemed as very useful. For research organizations and development projects in general, a common data format for both subjective and vehicular data would enable development of various tools on top of it, with clear efficiency advantages compared to the state-of-the-art.

A huge quantity of data (in the order of terabytes) has been processed, the size mostly depending on the amount of video cameras and selected resolution. Aggregated data has been loaded in the CDB (both in local private installations and in the shared cloud installation, that collects data from all the pilot sites). Scalability is necessary when dealing with such quantities of data. This is ever better supported by state-of-the-art cloud services, which platform was exploited by the CDB.

Algorithm development for the data toolchain took it seriously the time performance point, to ensure a fast processing of huge quantities of data. As an example, the execution time of vehicular data from three trips (total size of 78MB) by the CDB-aggregator takes 12 seconds on a lab PC. Input files, encoded in HDF5 format, contained the original vehicular signal timelines, enriched with the computed derived measures and driving scenarios. The Aggregator computed all the four types of CDB PIs (Trip PI, Scenario Specific Trip PI, Scenario Instance PI and Datapoints) and finally exported them to json, for direct uploading to the CDB, and to csv, for post-editing by the analysts.

6.4 Lessons Learnt

The piloting on real roads of SAE L3 ADFs [23] is a huge challenge, which required, in the L3Pilot case, the work of a consortium of 34 partners. The development of the shared data-flow toolchain was carried out by a team of researchers which addressed all the issues from data logging to the user interface to a shared database, which is now queried by data analysts to answer research questions on the impact of ADFs. The design and implementation tasks were iterative and the continuous interaction with members from both the methodology and experimental evaluation teams was crucial for tuning and validating requirements and for keeping the process in the right track. The overall team involved multiple people who were able to help each other facing the technological challenges and scientifically validate results. Vehicle owners played a key role in the development process, which was a big advantage for fully testing the tools on real automotive data and fully understanding the requirements. This also involved important aspects of confidentiality, property, and privacy, that had to be considered and adequately balanced.

Toolchain developers found it good practice to organize weekly scrum calls to review development status, list tasks and go through issues, according to an agile methodology. Weekly meetings also enabled creation of a well-functioning team across many organizations, and the use of a versioning tool such as GitLab was vital to manage the several teams working on complex hardware/software system development. We created in the project's backlog tasks for each one of the performance indicators, derived measures and driving scenarios that had to be developed. Task information was carefully updated and tracked. Every member would assign to himself the right tasks in the backlog. At the end of the implementation of each algorithm/module, one or more reviewers would be assigned for code review. Finally, a tester would be responsible for validating the new/updated algorithms on real data. A template for implementing and documenting functions was defined, so to target transparency and maintainability of the developed script. As an example, Figure 6-2 shows a commit pushed on GitLab for a driving scenario script.

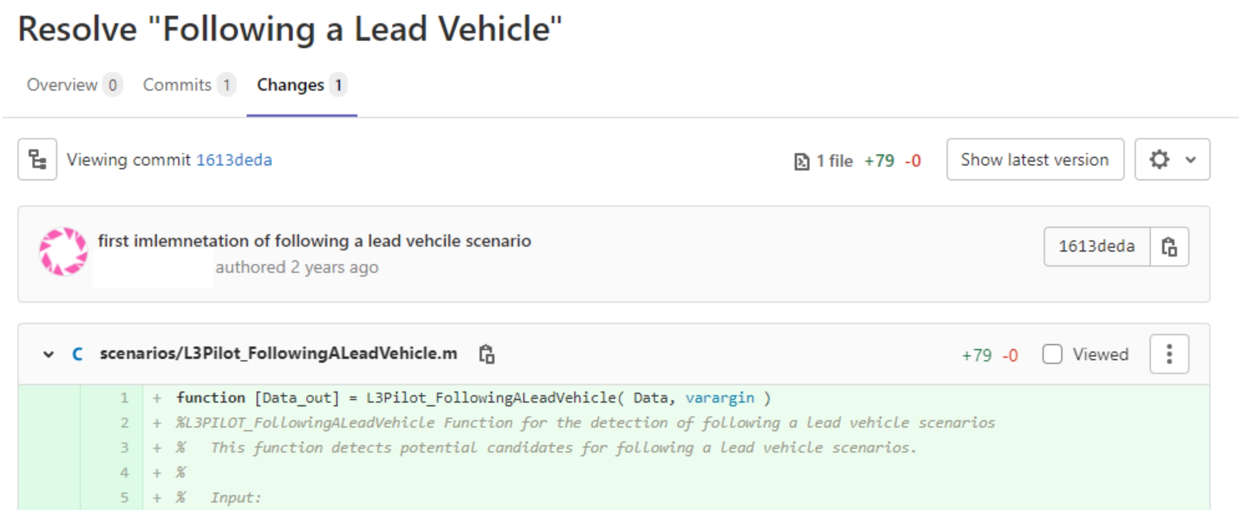

Figure 6-2 Example of a shared script on GitLab

While final results from the SP7 Evaluation teams are not available yet, the L3Pilot experience seems to indicate feasibility of assessing impact of novel ADFs exploiting multi-vehicle-owner data shared at the level of indicators and well-defined syntheses (e.g., the datapoints), which enables combining test results from several pilot sites, still keeping a good level of confidentiality. The scope of data processing scripts in a project such as L3Pilot has become very large. The number of derived measures, performance indicators and scenarios to be calculated out of test data amounts to hundreds. If pilot sites would each face such evaluation and calculation requirements alone, they would just end up carrying out a limited evaluation due to lack of resources. There are also general barriers that hinder data sharing, such as: intellectual property rights, privacy and product confidentiality, quality issues, lack of resources or trust, poor or missing agreements. This obviously stresses the importance of data sharing clauses in consortium agreements.

A key point, also for future research, concerns the study on how to improve CDF, particularly considering the different and sometimes conflicting needs of different users and stakeholders (e.g., different departments of an automotive company), also implying different types of source data, derived measures, detectable scenarios, and performance indicators.

Another aspect concerns the integration of all the modules in a single system, from converting data to the CDF up to sharing them on the CDB. Deploying the provided tools in the pilot sites sometimes required some level of IT knowledge. This was resolved through instructions and friendly collaboration. Combining user friendliness with leading edge research is not easy, but is a goal that could enhance effectiveness and efficiency in future projects.

While the implementation is exclusively in the automotive field, we argue that the proposed methodological approach and system architecture and tools are general and could be efficiently adapted and employed in different domains to support quantitative research analyses:

- The common data format can be defined for any application domain, if not yet available.
- The principles of the CDB-aggregator (segmentation and statistical data synthesis) are generally applicable. Different factors (experimental condition, types of context of usage of a new system to test, etc.) can be efficiently nested in the modular processing.
- The subjective data management tool is implemented for automotive driving, but its schema should be applicable to any investigation on human factors and user acceptance through questionnaires.

Finally, the data gathered through the project was large and advanced machine learning could be applied on these data, for instance to classify scenarios to help taking decisions while the system takes over the driving task.

7. Conclusions

Conducting a pilot on novel ADFs implies processing a huge amount of data, to extract meaningful information. To the best of our knowledge, there is no specific tool support for daily activities concerning data management in AD piloting, from logging to data synthesis to query for impact assessment RQs (considering requirements of confidentiality, data quality check, etc.). L3Pilot designed and implemented a confidentiality-aware toolchain to allow effective and efficient implementation of all the data-management related activities, by a variety of concerned actors.

L3Pilot created and promotes the common data format (CDF) for open collaboration in automotive research. The CDF allows homogenous processing of all the data across all the vehicle owners and pilot sites. It is a high-compression HDF5-based file format, where each trip is saved as one file, including metadata that provides further information of the recording.

The common format enabled the development of combined analysis scripts for all pilot sites that include indicator calculation, event and driving situation detection, and support for video data annotation. The shared calculation framework ensures that fully comparable indicators, distributions, and event lists can be extracted from each test. The resulting indicators and data distributions were further used as an input for impact assessment, where results from many pilot sites are statistically combined. The toolchain processing was applied to both objective (i.e., vehicular) and subjective (i.e., from user questionnaires) raw data.

As another key novelty of the project, a pseudonymized statistical aggregation of data from each test trip is stored in the CDB shared in the cloud to provide a quantitative basis for answering the set of 100+ research questions defined by L3Pilot. Combining results from different pilot sites enabled analysts to use a larger dataset for statistical work.

Establishing a collaborative community of researchers and developers who are knowledgeable in their respective domains was a key factor for the project. This team has been vital to allow a full understanding of the requirements, development of specifications and system and proper handling of all the issues that emerged with the concrete operations in the pilot sites. Discussions between experts in different fields have been very useful to achieve quality in a reasonable timeframe.

The toolchain was developed by a multidisciplinary team of partners, of which the candidate was part. Beside the general contribution to the overall system design, the candidate particularly focused on developing some modules, such as subjective data processing, and Performance Indicator computation (that represent the synthesis data sharable by all the vehicle owners in order to allow a quantitative response to the 100+ research questions defined by the L3Pilot analysts), and implementing some derived measures and driving scenarios, as detailed in this thesis.

The scientific value and originality of the candidate's contribution has been confirmed by the participation of the candidate in two articles that were published on high-quality peer-reviewed journals and three articles that were presented in peer-reviewed scientific conferences, as detailed at the end of this thesis.

Based on the project experience and feedback, we can indicate some directions for future research. These include extending and using the data processing and enrichment with DMs and driving scenarios module to answer further research questions. Another important point will be to optimize the MATLAB scripts in terms of memory occupation and time of execution. Also, the deployment of the scripts could be enhanced by exploiting docker installation. Another point concerns developing human computer interaction modules to enhance usability of the scripts.

Glossary of Acronyms and Abbreviations

The Glossary provides a list of key terms denoted in this document. The definitions are excerpt from previous work in the field of automated driving.

ADF	Automated Driving Function.
Automated Driving System	A combination of hardware and software required to realize an ADF [11].
Baseline	Set of data to which the performance and the effects of the technology under study are compared [25].
Derived Measures (DMs)	A single measure calculated from a direct measure (e.g. by applying mathematical or statistical operations) or a combination of one or more direct or derived measures [12].
Driving Scenarios	The abstraction and the general description of a driving situation without any specification of the parameters of the driving situation, thus, it summarizes a cluster of homogenous driving situations. Driving scenarios are typically short in time ($t < 30$ s) and only a few vehicles are involved. An example is lane change to the left lane [73].

Performance Indicator (PI)	Quantitative or qualitative indicator[s], derived from one or several measures, agreed on beforehand, expressed as a percentage, index, rate, frequencies, or other value, which are monitored at regular or irregular intervals and can be compared to one or more criteria. In some cases, these will be the same as a derived measure, in other cases, further processes are required to generate a PI [25].
Pilot Test	Field test of applications and functions not as mature as in FOTs. The methodology for testing, however, may be in principle the same. The test is used to decide how and whether to launch a full-scale project [25].
Raw Data	Data that has been recorded in instrumented vehicles (CAN data, video, GPS logs etc.). This data is by nature heterogeneous; different vehicles will produce different datasets. These datasets are thus not immediately useful for comparison [11].
Research Question (RQ)	A general question to be answered by compiling and testing related specific hypotheses [11].
SAE L3 – Conditional Automation	The driving mode-specific performance by an Automated Driving System of all aspects of the dynamic driving task with

	the expectation that the human driver will respond appropriately to a request to intervene [74,75].
SAE L4 – High Automation	The driving mode-specific performance by an Automated Driving System of all aspects of the dynamic driving task, even if a human driver does not respond appropriately to a request to intervene [75].
CDF	The common data format defined released open source to convert vehicular datasets to a common HDF5 structure.
CDB	A cloud database onto which objective and subjective synthesis indicators are uploaded from all the pilot sites to provide a quantitative basis for answering the project research questions.
Ego Vehicle	The self-driving vehicle in the road-test.
Lead Vehicle	The vehicle in front of the Ego Vehicle.
TTC	The time to collision.
THW	The time headway is the time difference between two successive vehicles as they cross the same point on the roadway [76].

FOT	Field Operational Test.
FESTA	Field opErational teSt supporT Action [12].

Bibliography

[1] *SMART Driverless golf cart provides a glimpse into a future of autonomous vehicles*. Available at https://news.mit.edu/2013/smart-driverless-golf-cart-provides-a-glimpse-into-a-future-of-autonomous-vehicles.

[2] *Induct Launches Navia, The First 100 Percent Electric, Automated Shuttle In The U.S.*. Available at http://www.prnewswire.com/news-releases/induct-launches-navia-the-first-100-percent-electric-automated-shuttle-in-the-us-238980311.html.

[3] *Ten ways autonomous driving could redefine the automotive world*. Available at http://www.mckinsey.com/industries/automotive-and-assembly/our-insights/ten-ways-autonomous-driving-could-redefine-the-automotive-world.

[4] S. Beiker, *Implementation of an automated mobility-on-demand system*, in *Autonomous Driving*, Springer, Berlin, Heidelberg, pp. 277–295.

[5] *Induct presents world's first fully-electric driverless shuttle: the Navia*. Available at http://induct-technology.com/en/files/2012/12/Navia-press-release.pdf.

[6] A. Bartels, M.M. Meinecke and S. Steinmeyer, *Lane Change Assistance*, in *Handbook of Driver Assistance Systems: Basic Information, Components and Systems for Active Safety and Comfort*, H. Winner, S. Hakuli, F. Lotz and C. Singer, eds., Springer International Publishing, Cham, 2016, pp. 1235–1257.

[7] S. Lüke, O. Fochler, T. Schaller and U. Regensburger, *Traffic Jam Assistance and Automation*, in *Handbook of Driver Assistance Systems: Basic Information, Components and Systems for Active Safety and Comfort*, H. Winner, S. Hakuli, F. Lotz and C. Singer, eds., Springer International Publishing, Cham, 2016, pp. 1287–1302.

[8] H. Winner and M. Schopper, *Adaptive Cruise Control*, in *Handbook of Driver Assistance Systems: Basic Information, Components and Systems for Active Safety and Comfort*, H. Winner, S. Hakuli, F. Lotz and C. Singer, eds., Springer International Publishing, Cham, 2016, pp. 1093–1148.

[9] PReVENT, *Code of Practice for Design and Evaluation of ADAS, version 5.0*, .

[10] H. Gotzig, *Parking Assistance*, in *Handbook of Driver Assistance Systems: Basic Information, Components and Systems for Active Safety and Comfort*, H. Winner, S. Hakuli, F. Lotz and C. Singer, eds., Springer International Publishing, Cham, 2016, pp. 1077–1092.

[11] D. Hibberd, T. Louw, E. Aittoniemi, R. Brouwer, M. Dotzauer, F. Fahrenkrog, S. Innamaa, S. Kuisma, N. Merat, B. Metz, N. Neila, M. Penttinen, P.P. Guillen, C. Rösener, A. Silla, T. Streubel, F. Tango, B. van den Boom, H. Weber et al., *L3Pilot Deliverable D3.1. From Research Questions to Logging Requirements*, (2018), https://l3pilot.eu/download/.

[12] FOT-Net and CARTRE, *FESTA Handbook Version 7*, (2018), https://connectedautomateddriving.eu/wp-content/uploads/2019/01/festa-handbook-version-7.pdf.

[13] *L3Pilot Common Data Format (L3Pilot CDF)*. Available at https://github.com/l3pilot/l3pilot-cdf.

[14] J. Hiller, S. Koskinen, R. Berta, N. Osman, B. Nagy, F. Bellotti, A. Rahman, E. Svanberg, H. Weber, E.H. Arnold, D. Mehdrad and A. De Gloria, *The L3Pilot data management toolchain for a level 3 vehicle automation pilot*, Electronics 9 (2020), .

[15] F. Bellotti, N. Osman, E.H. Arnold, S. Mozaffari, S. Innamaa, T. Louw, G. Torrao, H. Weber, J. Hiller, A. De Gloria, M. Dianati and R. Berta, *Managing Big Data for Addressing*

Research Driving Impact Assessment, Sensors 20 (2020), https://doi.org/10.3390/s20236773.

[16] B. Nagy, J. Hiller, N. Osman, S. Koskinen, E. Svanberg, F. Bellotti, R. Berta, A. Kobeissi and A. De Gloria, Building a Data Management Toolchain for a Level 3 Vehicle Automation Pilot', in Proceedings of the 26th ITS World Congress, 2019.

[17] J. Stapel, F.A. Mullakkal-Babu and R. Happee, *Automated driving reduces perceived workload, but monitoring causes higher cognitive load than manual driving*, Transportation research part F: traffic psychology and behaviour 60 (2019), .

[18] Z. Wang, Y. Wu and Q. Niu, *Multi-sensor fusion in automated driving: A survey*, IEEE Access 8 (2019), .

[19] M. Ardelt, C. Coester and N. Kaempchen, *Highly automated driving on freeways in real traffic using a probabilistic framework*, IEEE Transactions on Intelligent Transportation Systems 13 (2012), .

[20] G. Alessandretti, A. Amditis, S. Metzner, E. Johannson and F. Fahrenkrog, *InteractIVe Deliverable D1.9 - Final Report*, (2014), https://www.interactive-ip.eu/index.dhtml/docs/interactive_sp1_20140506v1.2-d19-final_report.pdf.

[21] *AdaptiVe Final project results*, (2017), https://www.adaptive-ip.eu/files/adaptive/content/downloads/adaptive-sp1-v1-0-dl-d1-0-final_report.pdf.

[22] I. Innamaa, S., Rämä, P., Hogema, J., Kroon, L., Heinig, I., Schindhelm, R., Ojeda, L., Servel, A., Visintainer, F., & Giosan, *Safety impacts of cooperative systems: DRIVE C2X Deliverables, Internal Report*, (2014), .

[23] SAE International, *Levels of driving automation*, SAE international (2018),

https://www.sae.org/news/2019/01/sae-updates-j3016-automated-driving-graphic.

[24] S. Innamaa, E. Aittoniemi, A. Bjorvatn, F. Fahrenkrog, J. Gwehenberger, E. Lehtonen, T. Louw, F. Malin, M. Penttinen, R. Schindhelm, A. Silla, H. Weber, M. Borrack, L. Di Lillo, N. Merat, B. Metz, Y. Page, E. Shi and H. Sintonen, *L3Pilot Deliverable D3.4. Evaluation Plan*, (2020), https://l3pilot.eu/download/.

[25] M. Penttinen, P. Rämä, M. Dotzauer, D. Hibberd, S. Innamaa, T. Louw, T. Streubel, B. Metz, J. Woerle, R. Brouwer, C. Rösener and H. Weber, *L3Pilot Deliverable D3.2. Experimental procedure*, (2019), https://l3pilot.eu/download/.

[26] H. Gimpel and M. Röglinger, *Digital transformation: changes and chances–insights based on an empirical study*, (2015), .

[27] C. Dremel, M.M. Herterich, J. Wulf, J.C. Waizmann and W. Brenner, *How AUDI AG established big data analytics in its digital transformation*, MIS Quarterly Executive 16 (2017), .

[28] J.R. Chandiramani, S. Bhandari and S.A. Hariprasad, Vehicle data acquisition and telemetry, in Proceedings of the 2014 Fifth International Conference on Signal and Image Processing, 2014, pp. 187–191.

[29] M. Benmimoun and A. Benmimoun, Large-scale FOT for analyzing the impacts of advanced driver assistance systems, in Proceedings of the 17th ITS World Congress, 2010.

[30] J. Pillmann, B. Sliwa and C. Wietfeld, The AutoMat CVIM—A Scalable Data Model for Automotive Big Data Marketplaces, in Proceedings of the 2018 19th IEEE International Conference on Mobile Data Management (MDM), , pp. 284–285.

[31] M. Johanson, S. Belenki, J. Jalminger, M. Fant and M. Gjertz, Big automotive data:

Leveraging large volumes of data for knowledge-driven product development, in Proceedings of the 2014 IEEE International Conference on Big Data (Big Data), 2014, pp. 736–741.

[32] R. Bartz, Mapping data models of the standardized automotive testing data storage to the unified modeling language, in Proceedings of the 2013 IEEE International Conference on Industrial Technology (ICIT), 2013, pp. 1327–1332.

[33] B. Schwab and T.H. Kolbe, Requirement analysis of 3D road space models for automated driving, in ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 4 (2019).

[34] A. Vu Vo, C.N.L. Hewage, G. Russo, N. Chauhan, D.F. Laefer, M. Bertolotto, N. Le-Khac and U. Oftendinger, Efficient LiDAR point cloud data encoding for scalable data management within the Hadoop eco-system, in Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), 2019, pp. 5644–5653.

[35] S. Biffl, T. Moser and D. Winkler, *Risk Assessment in Multi-Disciplinary (Software+) Engineering Projects*, International Journal of Software Engineering and Knowledge Engineering .

[36] J. Bhatia, T.D. Breaux, L. Friedberg, H. Hibshi and D. Smullen, Privacy Risk in Cybersecurity Data Sharing, in Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security, 2016, pp. 57–64.

[37] F. Fioretto and P. Van Hentenryck, Privacy-Preserving Federated Data Sharing, in Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS), 2019, pp. 638–646.

[38] Q. Hu, M. Cai, N. Mohabbati-Kalejahi, A. Mehdizadeh, M.A. Alamdar Yazdi, A. Vinel, S.E. Rigdon, K.C. Davis and F.M. Megahed, *A Review of Data Analytic Applications in Road Traffic Safety. Part 2: Prescriptive Modeling*, Sensors 20 (2020), https://www.mdpi.com/1424-8220/20/4/1096.

[39] V. Ratcheva, *Integrating diverse knowledge through boundary spanning processes—The case of multidisciplinary project teams*, International Journal of Project Management 27 (2009), .

[40] D. Winkler, F.J. Ekaputra, E. Serral and S. Biffl, Efficient data integration and communication issues in distributed engineering projects and project consortia, in Proceedings of the 14th International Conference on Knowledge Technologies and Data-driven Business, 2014, pp. 1–4.

[41] G. Burzio, G. Mussino, R. Tadei, G. Perboli, M. Dell'Amico and L. Guidotti, A subjective field test on lane departure warning function in the framework of the euroFOT project, in Proceedings of the 2nd Conference on Human System Interactions, 2009, pp. 608–610.

[42] Y. Barnard, S. Innamaa, S. Koskinen, H. Gellerman, E. Svanberg and H. Chen, *Methodology for Field Operational Tests of Automated Vehicles*, Transportation Research Procedia 14 (2016), .

[43] S. Innamaa, N. Merat, T. Louw, B. Metz, T. Streubel and C. Rösener, Methodological challenges related to real world automated driving pilots, in Proceedings of the 26th World Congress on Intelligent Transport Systems 2019, 2019.

[44] T. Louw, N. Merat, B. Metz, J. Wörle, G. Torrao and I. Satu, Assessing user behaviour and acceptance in real-world automated driving: The L3Pilot project approach, in Proceedings

of 8th Transport Research Arena TRA 2020, 2020.

[45] S. Nordhoff, T. Louw, S. Innamaa, E. Lehtonen, A. Beuster, G. Torrao, A. Bjorvatn, T. Kessel, F. Malin and R. Happee, *Using the UTAUT2 model to explain public acceptance of conditionally automated (L3) cars: A representative questionnaire study among 8,044 car drivers from seven European countries*, Transportation Research Part F: Traffic Psychology and Behaviour .

[46] Carmen Rodarius, J. Duflis, F. Fahrenkrog, C. Rösener, A. Várhelyi, R. Fernandez, L. Wang, P. Seiniger, D. Willemsen and L. van Rooij, *AdaptIVe Deliverable D7.1. Test and Evaluation Plan.*, (2016), https://knowledge-base.connectedautomateddriving.eu/wp-content/uploads/2020/04/data-sharing-framework-v1.1-final.pdf.

[47] M. Schulze, T. Mäkinen, T. Kessel, S. Metzner and H. Stoyanov, *DriveC2X Deliverable D11.6. Final Report*, 2014.

[48] M. Boban and P.M. D’Orey, Measurement-based evaluation of cooperative awareness for V2V and V2I communication, in Proceedings of the 2014 IEEE Vehicular Networking Conference (VNC), 2014, pp. 1–8.

[49] H. Gellerman, E. Svanberg and Y. Barnard, *Data Sharing of Transport Research Data*, Transportation Research Procedia 14 (2016), https://www.sciencedirect.com/science/article/pii/s2352146516302447.

[50] W. Wachenfeld, P. Junietz, R. Wenzel and H. Winner, The worst-time-to-collision metric for situation identification, in Proceedings of 2016 IEEE Intelligent Vehicles Symposium (IV), 2016, pp. 729–734.

[51] S. Hallerbach, Y. Xia, U. Eberle and F. Koester, *Simulation-based Identification of Critical*

Scenarios for Cooperative and Automated Vehicles, SAE International Journal of Connected and Automated Vehicles 1 (2018), .

[52] H.-P. Schoener and J. Mazzega, Introduction to PEGASUS, in China Autonomous Driving Testing Technology Innovation Conference, 2018.

[53] S. Ulbrich, T. Menzel, A. Reschka, F. Schuldt and M. Maurer, Defining and substantiating the terms scene, situation, and scenario for automated driving, in Proceedings of 2015 IEEE 18th International Conference on Intelligent Transportation Systems, 2015, pp. 982–988.

[54] H. Weber, J. Bock, J. Klimke, C. Roesener, J. Hiller, R. Krajewski, A. Zlocki and L. Eckstein, *A framework for definition of logical scenarios for safety assurance of automated driving*, Traffic Injury Prevention 20 (2019), .

[55] PEGASUS, *Basics for Testing - Stand 11*, https://www.pegasusprojekt.de/files/tmpl/pdf-symposium/11_scenario-formats.pdf.

[56] *OpenDRIVE*. Available at //www.asam.net/standards/detail/opendrive.

[57] *OpenSCENARIO*. Available at //www.asam.net/standards/detail/openscenario.

[58] Y. Barnard, H. Chen, S. Koskinen, S. Innamaa, H. Gellerman, E. Svanberg, A. Zlocki, C. Val, K. Quintero and D. Brizzolara, *D5. 4 Updated Version of the FESTA Handbook*, (2017), https://www.connectedautomateddriving.eu/wp-content/uploads/2017/08/2017-01-31_fot-net_festa-handbook.pdf.

[59] S. Innamaa, T. Louw, N. Merat, G. Torrao and E. Aittoniemi, Applying the FESTA methodology to automated driving pilots, in Proceedings of TRA2020, the 8th Transport Research Arena: Rethinking transport – towards clean and inclusive mobility, 2020, pp. 68.

[60] B. Metz, C. Rösener, E. Tyron Louw Aittoniemi, A. Bjorvatn, J. Wörle, H. Weber, G.A.

Torrao, A. Silla, S. Innamaa, F. Malin, E. Lehtonen, F. Fahrenkrog, P. Heum, K. Pedersen, N. Merat, S. Nordhoff, A. Beuster, M. Dotzauer and T. Streubel, *L3Pilot Deliverable D3.3. Evaluation methods*, (2019), https://l3pilot.eu/download/.

[61] W. Penard and T. van Werkhoven, *On the secure hash algorithm family*, Cryptography in context (2008), .

[62] *L3Pilot Deliverable D6.1. Pilot Execution Plan*, (2019), https://l3pilot.eu/download.

[63] B. Nagy, J. Hiller, E. Svanberg, M. Kremer, M. Luxen, F. Christen, S. Koskinen, F. Bellotti and N. Osman, *L3Pilot Deliverable D5.1. Pilot tools and data*, (2020), .

[64] J. Hiller, E. Svanberg, S. Koskinen, F. Bellotti and N. Osman, The L3Pilot Common Data Format—Enabling efficient automated driving data analysis, in Proceedings of the 26th International Technical Conference on the Enhanced Safety of Vehicles, Eindhoven, The Netherlands, 2019, pp. 10–13.

[65] *The HDF5 Library and File Format*. Available at https://www.hdfgroup.org/solutions/hdf5/.

[66] C. Gold, D. Damböck, L. Lorenz and K. Bengler, Take over! How long does it take to get the driver back into the loop?, in Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 2013, pp. 1938–1942.

[67] A. Breuer, J.A. Termöhlen, S. Homoceanu and T. Fingscheidt, OpenDD: A Large-Scale Roundabout Drone Dataset, in 2020 IEEE 23rd International Conference on Intelligent Transportation Systems, ITSC, 2020.

[68] L. Richardson, M. Amundsen and S. Ruby, *RESTful Web APIs'*, O'Reilly Media, Inc., Sevastopol, CA, USA, 2013.

[69] T. Steiner and J. Algermissen, Fulfilling the hypermedia constraint via http options, the http

vocabulary in rdf, and link headers, in Proceedings of the Second International Workshop on RESTful Design ACM, , pp. 11–14.

[70] F. Bellotti, R. Berta, A. Kobeissi, N. Osman, E. Arnold, M. Dianati, B. Nagy and A. De Gloria, Designing an IoT framework for automated driving impact analysis, in Proceedings of the 30th IEEE Intelligent Vehicles Symposium (IV), 2019, pp. 1111–1117.

[71] *LimeSurvey*. Available at https://www.limesurvey.org/.

[72] *Pattern (Java Platform SE 7)*. Available at https://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html.

[73] A. Bartels, U. Eberle and A. Knapp, *AdaptIVe Deliverable D2.1. System Classification and Glossary.*, (2015), https://www.adaptive-ip.eu/files/adaptive/content/downloads/deliverables & papers/adaptive-sp2-v12-dl-d2.1-system classification.pdf.

[74] SAE International, *Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems*, SAE Standard (2018), https://www.sae.org.

[75] P. WARRENDALE, *SAE International Releases Updated Visual Chart for Its "Levels of Driving Automation" Standard for Self-Driving Vehicles*, SAE Standard (2018), https://www.sae.org/news/press-room/2018/12/sae-international-releases-updated-visual-chart-for-its-"levels-of-driving-automation"-standard-for-self-driving-vehicles.

[76] S. Das, A.K. Maurya and A.K. Budhkar, *Determinants of time headway in staggered car-following conditions*, Transportation Letters 11 (2019), .

Published Articles Co-Authored by the Candidate

The results in this thesis have been published in several journals and conferences as listed in the following.

Journal Papers

1. Bellotti, F., **Osman, N.**, Arnold, E.H., Mozaffari, S., Innamaa, S., Louw, T., Torrao, G., Weber, H., Hiller, J., De Gloria, A., Dianati, M., and Berta R., 2020. Managing Big Data for Addressing Research Questions in a Collaborative Project on Automated Driving Impact Assessment. **Sensors**, 20(23), p.6773.
2. Hiller, J., Koskinen, S., Berta, R., **Osman, N.**, Nagy, B., Bellotti, F., Rahman, A., Svanberg, E., Weber, H., Arnold, E.H. Dianati, M., and De Gloria A., 2020. The L3Pilot data management toolchain for a level 3 vehicle automation pilot. **Electronics**, 9(5), p.809.

Conference Papers

1. Nagy, B., Hiller, J., **Osman, N.**, Koskinen, S., Svanberg, E., Bellotti, F., Berta, R., Kobeissi, A. and De Gloria, A., 2019, October. Building a Data Management Toolchain for a Level 3 Vehicle Automation Pilot. In 26th **Intelligent Transport Systems World Congress**, ITS Singapore 2019.

2. Bellotti, F., Berta, R., Kobeissi, A., **Osman, N.**, Arnold, E., Dianati, M., Nagy, B. and De Gloria, A., 2019, June. Designing an IoT framework for automated driving impact analysis. In 2019 **IEEE Intelligent Vehicles Symposium** (IV) (pp. 1111-1117). IEEE.
3. Hiller, J., Svanberg, E., Koskinen, S., Bellotti, F. and **Osman, N.**, 2019, June. The L3Pilot Common Data Format—Enabling efficient automated driving data analysis. In Proceedings of the 26th **International Technical Conference on the Enhanced Safety of Vehicles**, Eindhoven, The Netherlands (pp. 10-13).

Posters

1. **Osman, N.**, Bellotti, F., Berta, R., & De Gloria, A. 2020, October. Data Segmentation on Driving Scenarios for L3 Pilot. **In L3Piltot Summer School 2020**.