CLOUDIA

Methods for efficient search of events in large volumes of high-resolution data for the development of ADAS

Public report

Picture courtesy of Volvo Group

Project within FFI Safe Automated Driving

Author Fredrik von Corswant (main author and editor). Co-authors: Annika Larsson, Marcus Ryman, Iman Shahmari, Christian Berger, Ola Benderius, Maria Klingegård and Oliver Brunnegård

Date 2024-10-29

Fordonsstrategisk Forskning och Innovation

Jan 2019

Content

FFI in short

FFI is a partnership between the Swedish government and automotive industry for joint funding of research, innovation and development concentrating on Climate & Environment and Safety. FFI has R&D activities worth approx. €100 million per year, of which about €40 is governmental funding.

For more information: www.vinnova.se/ffi

1 Summary

Data collected from test vehicles during realistic driving conditions is important for the development and validation of new vehicle functions, for example advanced driver support systems (ADAS) and autonomous driving (AD). In particular, the increased use of machine learning and the vast variety of traffic situations to be covered increases the need for data. When data has been collected, the next challenge is to actually find the specific data that is of relevance for the development and validation activities to be performed. For this type of development, specific traffic scenarios, often referred to as *events*, are of specific interest. Developers may be interested in events like "a car stopped at a zebra crossing when a bicycle passed" or "a car left the highway on an exit ramp". Finding this kind of specific events in large amounts of unstructured driving data may be like finding a needle in a haystack. This project has therefore been focused on developing a method for defining specific traffic events, aiming at making such events efficiently searchable in large amounts of high-resolution naturalistic driving data that has been collected from a large number of vehicles in the field.

The project work was divided into four main work packages. First, based on a review of some previous projects and project workshops, a method was developed for defining complex traffic events and decomposing them into *event components*. Within the event, the different actors, actions, objects and conditions are identified. Secondly, to enable collection of driving data to be used in the project, a data collection platform used in previous projects was upgraded and installed in three different test vehicles. In this way data could be collected under both controlled and uncontrolled conditions. The data collection platform gathers data from several sensors in the test vehicle (cameras, GPS, IMU, driver monitoring system) and uploads it to a database on a cloud server. Even if only three test vehicles were available for this project, it was important to verify that the system can handle a large number of connected vehicles. By using simulation techniques, it could be shown that the system can handle more than one million connected vehicles simultaneously.

With the event method as a basis, the third work package focused on developing a search function enabling users to search for specific events in the database with driving data. Users can easily describe an event as plain text. Natural Language Processing techniques (NLP) are then used for breaking the event down into event components. As an alternative, users can define an event using graphical block diagrams. For users more used to coding, an application programming interface (API) can also be provided. Different algorithms, referred to as *detectors,* are used by the system for finding the different event components in the database. The project has, for example, looked into some kernel-based triggers, and the use of machine learning for finding objects in video streams, to function as detectors. Though, it has not been the focus of this project to develop a comprehensive set of detectors. The search function can also manage *time links* between different event components, distinguishing between things happening simultaneously or sequentially during an event. The user is then presented with a list of all events found in the database and, by clicking on each event, more detailed information (including the raw data from different sensors) can be accessed and downloaded. Algorithms for narrowing the search,

based on what event results the user likes and not, were also implemented using dynamic time warping techniques.

The fourth work package focused on demonstrating the possibilities with the event search function. The system has been demonstrated to different teams within the project partner organizations as well as to other researchers within academia and industry. The project was also presented at Transportforum 2024.

Taken together, the project has fulfilled the objectives set out in the application, providing valuable knowledge on how to define and search for specific traffic events. While some further development is still needed, the ambition is to bring the search function towards a commercial product, in collaboration with some automotive companies. While the project has mainly been focused on the automotive sector, and specifically ADAS, there is a potential use for this type of data search tools also in other lines of business including, for example, traffic insurance and community and traffic planning.

The project has been conducted in close collaboration between Pionate AB, Magna Electronics Sweden AB (initially Veoneer Sweden AB), Qualcomm Technologies Sweden AB (initially Veoneer Sweden AB), and Folksam Ömsesidig Sakförsäkring. The project started in November 2021 and ended in October 2024. The total project budget has been SEK 4.1 million. The project has received SEK 2.0 million funding from the FFI program Safe Automated Driving that made this project possible and that we are very grateful for.

2 Sammanfattning på Svenska

Utvecklingen av nya avancerade förarstödssystem och självkörandefunktioner i både personbilar och tunga fordon gör att det finns ett stort behov av data från körning i verkliga trafiksituationer, som kan användas för att träna och verifiera sådana system. Genom att utrusta testfordon som kör runt i trafiken med en rad olika sensorer såsom kameror, radar, lidar, GPS, IMU (accelerometrar) så kan information om olika händelser och skeenden i trafiken samlas in. Sådana händelser brukar med en engelsk term kallas ”event”. Med högupplösta sensorer så skapas varje sekund många gigabyte data. Totalt kan det bli flera tusen timmars körning, så det är mycket stora datamängder som behöver hanteras. Detta ger i sin tur utmaningar gällande insamlingen av data från fordonen. Så stora mängder data kan inte laddas upp via trådlös uppkoppling i mobilnätet. I stället får testfordonen komma ”hem” för byte av lagringsmedia eller så används logistikpartners som åker runt och byter ut lagringsmedia i fordonen. Vidare kan det vara utmanande att finna exakt de situationer och händelser (event) i den stora mängden data som är relevanta för en specifik utvecklare eller användare. Det blir som att söka efter en nål i en höstack.

Med detta som utgångspunkt har syftet med projektet CLOUDIA varit att utveckla och demonstrera en metodik för att definiera specifika event och göra dessa effektivt sökbara i stora mängder högupplöst data som har samlats in från ett stort antal uppkopplade fordon. Utifrån syftet formulerades följande två övergripande forskningsfrågor (RQ):

- RQ1: Hur kan högupplöst data effektivt samlas in från ett stort antal uppkopplade fordon?
- RQ2: Hur kan sökning efter specifika event göras effektivt i stora mängder insamlade data?

Den första frågan fokuserar dels hur insamling av data kan göras mer effektiv, dels hur skalbar insamling av data från uppkopplade fordon kan göras för att hantera ett mycket stort antal uppkopplade fordon. Den andra frågan fokuserar på sökning efter specifika event i stora mängder data, vilket i sin tur skapar behov av att utveckla en metodik för att definiera event.

Med utgångspunkt i projektets syfte och forskningsfrågorna definierades fyra hörnstenar, eller arbetspaket, för projektet:

- Utveckling av metodik för att definiera event. Genom att på ett strukturerat sätt bryta ner komplexa event i mindre beståndsdelar kan dessa event göras mer effektivt sökbara.
- Uppgradera plattform för insamling av data från fordon och testa skalbarheten av systemet med simuleringar. En plattform för insamling av data som har använts i tidigare projekt behövde uppgraderas avseende mjukvara, anslutning av ny sensor samt göra den enklare att installera i fordon. Då det endast finns utrymme att bygga ett begränsat antal testfordon i projektet så simuleras i stället uppkoppling av ett stort antal fordon.

- Implementering av sökfunktion för effektiv sökning efter specifika event i stora datamängder. I det här arbetspaketet utvecklas dels ”backend” i form av databas(er) för strukturerad lagring av data som har samlats in från fordonen, dels ”frontend” i form av ett användarvänligt gränssnitt för att göra sökningar efter specifika event.
- Demonstrera forskningsplattformens möjligheter för andra forskare och potentiella användare av systemet.

Ett flertal olika metoder och tillvägagångssätt har tillämpats under projektet. Utvecklingen av metodiken för att definiera event gjordes huvudsakligen genom workshops i projektteamet, men även baserat på en litteraturstudie av tre EU-projekt som haft liknande fokus. För insamling av egna data från fordon byggdes totalt tre olika testfordon som utrustades med den uppgraderade plattformen för dataloggning. Utöver egna data, användes även annan kördata från publikt tillgängliga dataset. Flera workshops arrangerades för att testa och utvärdera systemet för sökning av event och videor har gjorts för att kommunicera resultaten. Projektet har även presenterats för fler medarbetare inom projektets partnerorganisationer samt presenterats på en projektdag och en forskarkonferens.

Projektet har utvecklat en metodik för att dela in mer eller mindre komplexa event i mindre komponenter som på så sätt var och en kan göras sökbara. Detta medger också att olika sådana eventkomponenter enkelt kan kombineras till nya specifika event. Eventen delas in i aktörer, aktiviteter, objekt och egenskaper. Aktörer är de som utför aktiviteter. Objekt är föremål som i sig inte är aktiva men som är relevanta för eventet, exempelvis en stoppskylt. Egenskaper används för att beskriva villkor som är konstanta under ett event, exempelvis ”soligt väder” eller ”motorväg”. Exempelvis kan ett event beskrivas som ”en bil passerar en varningsskylt och kör sedan om en lastbil på en landsväg i regnigt väder”. Genom att använda språkmodeller (Natural Language Processing, NLP) kan eventbeskrivningen brytas ner i sina beståndsdelar (aktörer, aktiviteter osv). Då erhålls även information om inbördes förhållanden mellan eventets olika delar, exempelvis att bilen *först* passerar en varningsskylt *sen* kör om lastbilen.

För att göra sökningen efter specifika event så enkel som möjligt har en sökfunktion baserad på Natural Language Processing (NLP) utvecklats (”frontend”). Det innebär att det eftersökta eventet kan beskrivas av användaren i klartext och att systemet sedan automatiskt översätter texten till sökbara termer. Systemet erbjuder även ett grafiskt användargränssnitt där ett event kan byggas upp genom att länka samman grafiska objekt. För mer avancerade användare finns också ett kommunikationsgränssnitt (API) för programmering av eventsökningar.

Data som samlas in från testfordon lagras i en databas. Vid sökning efter specifika eventkomponenter används så kallade *detektorer* (speciella algoritmer) för att hitta förekomster av just denna eventkomponent i databasen. En form av detektorer kallas triggers och dessa baseras på matematisk analys av specifika signaler. Detta gör att man kan detektera exempelvis hårda inbromsningar eller filbyten baserat på data från accelerometrar. Vidare baseras detektorer på exempelvis maskininlärning för att exempelvis detektera specifika objekt i en videoström. Även om flera detektorer har utvecklats och använts inom projektet så har syftet inte varit att utveckla en heltäckande uppsättning detektorer för alla typer av eventkomponenter. Efterhand som data analyseras med hjälp av olika detektorer så taggas data så att det nästa gång går mycket snabbare att söka igenom samma mängd data.

Uppbyggnaden av databasen medger också att (ostrukturerad) data från andra dataset kan matas in och analyseras. I projektet har data från bland annat KITTI och Zenuity Open Dataset (ZOD) använts dels för att demonstrera systemets flexibilitet, dels för att testa skalbarhet i termer av att hantera stora mängder data. Vidare medger den flexibla databasstrukturen att data kan delas mellan olika användare (i den utsträckning detta är önskvärt och tillämpligt med avseende på exempelvis GDPR). Det finns också möjlighet för enskilda användare att utveckla sina egna detektorer och även dessa kan delas med andra aktörer. På så sätt kan insamlad data få ett bredare användningsområde och olika användare kan hjälpa varandra med exempelvis verifiering av vissa funktioner.

För utvecklingen av ett system för sökning efter event behövdes data från verklig körning, både så kallad naturalistisk data från körning på allmän väg och data som kommer från tester som görs under mer kontrollerade former på en provbana. För ändamålet vidareutvecklades den plattform för datainsamling som använts i två tidigare forskningsprojekt. Den består av en dator med ett 4G-modem och anslutningar för både ethernet och CAN, en GNSS-enhet som även har accelerometrar och gyro inbyggda, en videoserver som kan hantera upp till fyra kameror (tre användes i projektet), samt ett system för att studera förarens blickbeteende. Mjukvaran uppgraderades för att göra insamlingen av data effektivare och ytterligare en kamera som också kan ge objektinformation kopplades in. Plattformen anpassades också för att enklare kunna installeras i fordon. Totalt installerades plattformen för loggning av data i tre olika fordon.

Då utvecklingen av ADAS och liknande system kräver stora mängder data från ett stort antal körfall är det viktigt att kunna koppla upp och samla in data från ett stort antal fordon samtidigt. Inom ramen för projektet fanns det inte möjlighet att bygga ett stort antal testfordon. I stället användes virtuella servrar på Amazon Cloud där data från testfordon laddades upp och på så sätt kunde användas för att simulera att ett stort antal fordon samtidigt kopplar upp sig mot databasen. Som mest simulerades över en miljon uppkopplade fordon, vilket tydligt visar på systemet skalbarhet.

Sammantaget har projektet varit framgångsrikt då det har bidragit till både kunskapsuppbyggnad och utbyte av erfarenheter mellan de medverkande företagen. Projektet har också bidragit till flera av FFI:s övergripande mål och specifikt till målen för delprogrammet Trafiksäker automatisering. Projektets resultat har presenterats internt hos de medverkande företagen och på en forskarkonferens. Planen är nu att försöka ta nästa steg och göra systemet för sökning av event till en kommersiell produkt.

3 Background

Traditionally, crash data has been used to focus the efforts in developing safer cars. With Advanced Driver Assist Systems (ADAS), such as adaptive cruise control (ACC), data from crashes is no longer sufficient (Veoneer and SwissRe, 2021). As ADAS develops from more standalone functions, such as blind spot detection, towards automation of more integrated functions like lane keeping support (LKS), more advanced methods and tools, often using machine learning, plays a in increasingly important role. At the same time, researchers and developers need to study and understand how the human driver perceives different situations and what actions are taken, including how ADAS is used in different situations. Taken together, this development drives a need for collecting large amounts of high-resolution naturalistic driving data from large numbers of vehicles, both test vehicles and vehicles in regular traffic.

Collecting large amounts of data from a large number of vehicles is a challenge. Traditionally, much of the data that has been collected has relatively low resolution (for example vehicle position and speed) and therefore generates relatively small amounts of data (typically some megabits per second). Also, data from more advanced sensors like cameras or radars, that is interpreted as objects (for example ”a car is detected at X meters distance) generates relatively limited amounts of data. When data volumes are small and there is no need for uploading data in real time, connected systems based on, for example, 4G can be used.

However, when data is collected for e.g. training of AI/ML systems, often unprocessed data with high resolution is needed. Cameras, LiDAR and radar are examples of sensors that generate large amounts of data and, with several such sensors in a vehicle, the amount of data generated can surmount to around 80 Gbit/second. Thus, many times more than what can be handled with a 4G connection. The large amounts of data generated therefore makes it very challenging to upload all data from vehicles to a server via wireless networks. Therefore, most available solutions rely on collection of such high-resolution data that is stored locally in the vehicle on hard drives or solid state drives. This, in turn, leads to relatively high costs for collecting data from vehicles on the field. Often logistics companies are used for swapping disks during nights when vehicles are not used. The cost for collecting the data also increases with the number of vehicles, thus limiting the scalability. It can also take long time, often several days or weeks from data is created in a vehicle until it is made available for those who need it. Therefore, there is a need for solutions providing faster and more efficient collection of such high-resolution data.

When the data from vehicles is made available to users in large databases, the next challenge is to find what they are looking for. The amount of data is very large and just using simple search criteria like vehicle acceleration, speed and position will not suffice. Instead, there is a need for making advanced searches for specific complex traffic scenarios and sequences of specific activities performed by certain actors, so called *events*. This includes, for example, lane changes, driving at intersections, interaction with other road users, or near crash situations. Also, events involving driver behavior (e.g. driver is looking at the cell phone) can be of much interest. Thus, there is a need for developing methods for clearly defining events combined with the development of more advanced search functions that make it possible to find specific events in large amounts of data.

4 Purpose, research questions and method

4.1 Purpose

The overall purpose with the project has been to develop and demonstrate methods for defining specific events and make these efficiently searchable in large amounts of high resolution data that has been collected from a large number of connected vehicles.

By collecting high-resolution naturalistic data and making the data efficiently searchable it can be made useful for research on and development of, for example, ADAS.

4.2 Research questions

The following two research questions were set out already in the application for the project.

- RQ1: How can high-resolution data be efficiently collected from a large number of connected vehicles?
- RQ2: How can the search for specific events be performed efficiently in large amounts of available data?

The first research question (RQ1) aimed at investigating how data collection can be made more efficient in the sense that larger amounts of high-resolution data can be uploaded from vehicles, as an alternative to storing the data locally in the vehicle. It also aimed at investigating how data collection can be scaled up in terms of number of vehicles connected. The second research question (RQ2) aimed at addressing how to make specific events searchable. This in turn required that a method for defining such events was defined.

4.3 Project cornerstones

Based on the project purpose and the research questions, four cornerstones for the project were defined, see Figure 1. As a first step, based on workshops, studies of previous projects as well as analytical work, a method for defining events was developed. Secondly, the data collection platform developed in a previous project was upgraded in order to make data collection even more efficient. The data collection platform was installed in three different vehicles for collection of naturalistic driving data. Moreover, the scalability of the data collection system was tested through simulation of data flows. Thirdly, with the event definition method as a basis, a user-friendly search function for finding specific events in large amounts of high-resolution data was developed. As a last step, the search function was demonstrated for project members and other interested parties and an illustrating video was created. More details on the outcomes of each of these project corner stones are presented in the following chapters.

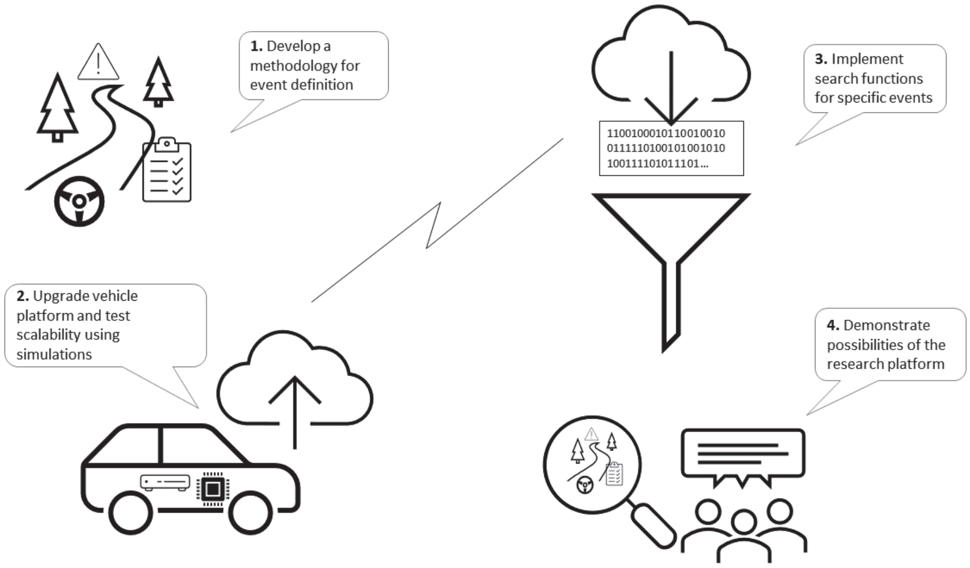

Figure 1. Illustration of the four project cornerstones.

The project was carried out in close collaboration between Folksam who among other things perform research on ADAS connected to accident statistics, Veoneer who develops advanced vehicle safety systems (e.g. sensor systems detecting driver behavior), and Pionate who are specializing in collection and analysis of large amounts of high-resolution data from connected vehicles. During the course of the project, Veoneer was split into two companies, eventually resulting in two ”new” project partners, Magna and Qualcomm.

4.4 Method

Performing the project involved a mix of theoretical as well as practical methods. This enabled several iterations where the solutions developed in the project could be tested and evaluated, while providing valuable feedback and information.

In order to develop a method for defining events a project workshop was held to discuss the needs when studying ADAS effects on normal driving. Participants were representatives from Pionate, Veoneer (later Magna and Qualcomm), and Folksam. This was important to include all project partners' perspectives and reach a common view on what to be achieved. Also, methods applied, and results obtained in three other (large EU projects) were studied as input and reference. Later, when some initial concepts had been developed, these were presented, discussed and refined in more workshops. In parallel, the development of the more hands-on search function had started, which also provided valuable feedback for the event definition method discussions.

In order to enable testing and evaluation of the event definition method, as well as the system for data collection and event searching, the access to high-resolution naturalistic driving data was important. Therefore, a considerable amount of effort was put into building and installing data collection systems in three different test vehicles. As a first step, an evaluation system was installed in one vehicle by Pionate. After some further refinements, two more (slightly upgraded) systems were installed in test vehicles by Magna and Qualcomm respectively. This setup enabled controlled collection of data for the project, where very specific pre-defined events could be recorded. For that purpose, the airfield in Vårgårda was used as a test track with no other public traffic present. Also, naturalistic data was collected with these vehicles by driving on public roads.

Further, to demonstrate the flexibility and scalability of the event searching system in terms of its ability to handle different types of driving data, as well as handling large amounts of data, naturalistic driving data from other datasets such as KITTI and Zenuity Open Dataset (ZOD) was included.

In this project, it was not possible to install data collection systems in thousands or hundreds of thousand vehicles. Therefore, to test and evaluate the scalability of the data collection system, different simulation methods were applied. By utilizing Amazon Cloud virtual previously recorded naturalistic driving data could be stored on virtual servers acting as a large number of connected vehicles. By connecting these servers to the cloud server used for the project, simultaneous data collection from a large number of vehicles could be simulated. Thereby the system could be stress-tested in a relatively fast and economically viable way.

For the development of the event search function, a combination of software development, workshops and demonstrations was used in several iterations. The different parts of the search function, for example the Natural Language Programming (NLP) enabling free text search, were first developed as a stand-alone module that could be evaluated by the rest of the project team. The access to naturalistic driving data collected within the project also helped the development of the event search functions. When the different event search modules could be integrated to a complete system, it was demonstrated "hands on" for all project members. Also, videos demonstrating the system were made. The search system was also demonstrated to a somewhat wider audience within the project partners.

5 Objective

The overall project objective is to make the collection of high-resolution naturalistic driving data more efficient, make specific traffic events efficiently searchable and, thereby, more useful for the research and development of ADAS functions including drivers' use of ADAS. In the project application the following project objectives were outlined.

- Develop a method for defining complex events that enables efficient search in large amounts of data. The method shall be made publicly available to other actors.
- Develop a compact, robust and scalable platform for efficient collection of high-resolution data in vehicles.
- Reduce the time from the moment data is created in a vehicle until information about an event is available for developers in the order from days to seconds.
- Implement a search function in the cloud environment enabling advanced search for events.
- Demonstrate how selected events can be efficiently searched for in large amounts of collected data.
- Demonstrate the scalability of the system, by using simulations, showing that 1000-100,000 vehicles can be connected simultaneously.

The objectives have been kept more or less unchanged throughout the project and have by and large been fulfilled by the project. When it comes to reducing the time it takes for making data available for users (item #3 in the list above), the project has demonstrated how this can be done, but the full functionality for accessing data more or less instantaneously from vehicle logging platforms has not yet been implemented.

6 Results and deliverables

As set out in the project purpose and objectives, a methodology for defining events has been developed enabling the study of ADAS in normal day-to-day driving. A vehicle data logging platform has been further refined and built in to test vehicles enabling collection of naturalistic driving data. Furthermore, with the event definition method as a basis and aimed at using collected driving data, a cloud-based search function has been developed. This enables searching for specific events in large amounts of unstructured data. Also, the scalability of the cloud-based system has been tested through simulations. In the following sections, the project results and deliverables are described in more detail.

6.1 Method for defining events

To enable searching for specific events in large amounts of unstructured data, events must be defined in a structured way. As a first step, a literature review was performed to collect knowledge on what had been done in previous projects. Based on some inputs from the literature survey and a series of workshops in the project team, a conceptual method was developed where events are divided into event components that can be searched for by defining specific event triggers. This is described in more detail in the following sections (see Larsson & Ryman, 2023 for more details). It makes the basis for the database and search functions that are explained in more detail later on.

6.1.1 Literature review

As an initial step, to understand how driving behavior and driving situations have previously been categorized, a literature review was performed. The review included the EuroFOT, PEGASUS, and L3Pilot projects as examples of large-scale data collections with a focus on scenarios and events in relation to assisted and/or automated driving.

EuroFOT

EuroFOT was the first large-scale EU project to evaluate Active Safety systems in field tests. The project was conducted from 2008 to 2012 and included the collection and analysis of road data from 35 million kilometers drive by 1200 drivers in 4 countries (Kessler et al., 2012).

Two definitions are primarily used within EuroFOT to categorize and describe factors that impact driving: *situations* and *events* (Faber et al., 2011). A situation is defined by a collection of 'situational variables' - characteristics of the traffic situation that is relevant for the vehicle. For example: functional status for the vehicle (available ADAS systems), speed limit, number of road lanes etc. An event is defined as something that takes place within a given time period and is described as a combination of observations according to predefined rules. EuroFOT only considers events to be relevant for security analysis. Examples include lane change, critical distance, crash and so on.

A couple of methodological choices need to be noted for the categorization of factors relevant for *driving* in EuroFOT. First, events can be of varying complexity. They can differ both in abstract and conceptual description in ways that could impact data extraction. It is also possible that some driving could fail to be included as either an event or a situation per the EuroFOT categorization, such as vehicle following, driving with a tired or distracted driver etc. Such examples show driving circumstances that could be of interest, but that are not bound to local time horizons but span longer periods and therefore would not be classified as events. It might be possible to extend the definition of situations to include such types of driving, or we might require a new category all together.

PEGASUS

The purpose of the PEGASUS project (Pegasus, 2019) was to develop a standardized process for testing autonomous drive functions. The project was a collaboration between industry and academia (primarily in Germany), resulting in a number of publications, symposia and the so-called PEGASUS method, which included specifications for verification and validation of autonomous drive functions and describe relevant scenarios.

In PEGASUS, driving is described through the lens of Scenarios. A model based on six layers is used to describe different parts relevant for a scenario.

Layer 1 Road-level	Geometry, topology, boundaries, etc.
Layer 2 Traffic Infrastructure,	Traffic signs, elevated barriers, etc.
Layer 3: Temporary manipulation of layer 1 and 2	Road work, etc.
Layer 4: Object	Static, dynamic, movable, maneuvers, etc.
Layer 5: Environments	Weather, lighting, and surrounding conditions
Layer 6: Digital information	V2X, digital maps, etc.

The PEGASUS method also divides scenarios into the three categories functional, logical and concrete, depending on its level of abstraction. For functional scenarios*,* all six layers are described in natural language. For logical scenarios*,* the layers are described with intervals for permissible parameter values. Concrete scenarios are specific instantiations of a logical scenario.

To generate concrete scenarios from field data using the PEGASUS method, it is necessary to create a scenario catalogue for the fourth layer in the 6-layer model. This scenario catalogue is formulated from the point of view of safety relevant scenarios, defined from the point of view of potential collisions with the ego vehicle (i.e. the vehicle logging the data) and some other object. The logical scenarios are then specified based on where the collision takes place: at the front, back, or side of the ego vehicle, as well as the initial location of the other object: in front, behind or to the side of the ego vehicle. This way, there are nine possible logical scenarios.

One important aspect to note for the PEGASUS project is that its purpose is to standardize testing of autonomous drive functions. The method is therefore developed for representation of generic scenarios that enable testing through simulation and on test tracks.

The main purpose has not been to formulate a structured categorization of *driving,* which, as a result, does not fully fit the purpose in the CLOUDIA project.

The nine logical scenarios in PEGASUS describe only a limited number of scenarios of potential interest. It may however be of value to study the contents of the layered model and the data formats especially for inspiring formulation of information relevant for describing a generic event, even though not all information would be necessary for our purposes.

L3Pilot

The third investigated project is L3Pilot, a large-scale European research project intended for the evaluation of the safety and effectiveness of advanced autonomous drive functions (Etemad, 2021). Drive functions are evaluated in simulation and with real road data from 1000 drivers in 100 cars in over 10 countries. The categorization of *driving* in L3Pilot was done by considering *driving scenarios* (Metz et al., 2019; Weber et al., 2021). Such a scenario describes the development of a situation in a traffic context where at least one actor performs a specified action and/or are influenced by an event. The action or the event is defined without concrete parameters and the influenced actor could be the ego vehicle or another actor in the scenario. Examples include 'independent driving', 'approaching lead vehicle' and 'lane change'. A pre-defined catalogue of drive scenarios is created to contain scenarios that are intended to be mutually exclusive and exhaustive. All analysis performed within L3Pilot is presented individually for each drive scenario. The motivation for this presentation is that naturalistic drive studies contain many uncontrolled variables. Separating the analysis into multiple drive scenarios is an attempt to minimize the variability of such variables.

The method of categorization of *driving* in L3Pilot is interesting but suffers from significant shortcomings for use in CLOUDIA. The relatively few scenarios that are described are quite general so there are still many uncontrolled variables that can cause two situational instances of the same scenario to be widely different. Variables of interest, such as driver state, external environment, and other vehicles and objects, are left out of the definition of scenarios. The benefit of the limited and generic scenario descriptions is the ease of extracting relevant information from data.

6.1.2 Descriptions and categorization to describe driving

The literature review has proven the importance of the intended use as a departure point for describing and categorizing driving. In the CLOUDIA project, the goal is to create a method for specifying a particular type of driving situation, event or a scenario of interest and to be able to extract it from a high-resolution database. Therefore, to determine a suitable method for the intended use in CLOUDIA, the methods found through the literature study should be evaluated from the point of view of the project's intended objectives. In CLOUDIA, the research need is to extract relevant driving data from naturalistic driving, requiring the description of many types of driving scenes as well as the differences between them.

The three projects studied represent different views on how driving can be described and categorized, due to their different objectives. Thus, the optimal solution will depend on the intended goal of the analysis. It would not be feasible for a large scenario database to be exhaustive and contain only scenarios that are mutually exclusive. As the complexity of the situations increase in real-life data, driving scenarios cannot be defined by parameter intervals alone without iteration. A flexible approach is needed. An overtaking scenario should be possible to classify as such also in a scenario when parameters fall outside of previously defined boundaries.

A benefit of the approach employed in EuroFOT is that the categorization into Situations can decrease the variability of driving scenes without a corresponding decrease in the generality of the definition of Events. We therefore suggest basing the approach to classification in CLOUDIA on the methodology employed in EuroFOT.

6.1.3 CLOUDIA event definitions

Based on learnings from the literature review as well as discussions within the project team during workshops, the following terminology for defining *events* is suggested to suit the specific CLOUDIA project needs:

An event contains one or more actors, performing one or several actions in relation to or impacting objects or other actors, during a specific time period.

Let’s have a look at an example (see Figure 2). A simple event can be defined as ”a car is overtaking a truck”. In this example, the main actor is the car performing the action ”overtaking”. The truck is in this case regarded as an object. Also, the time period for the duration of the event can be used to specify the event. For example, ”a car is overtaking a truck during less than 25 seconds”.

Figure 2. Event example: A car overtaking a truck

The event can also be defined in more detail by, for example, adding more objects: ”a car passes a speed sign when overtaking a truck”. In this case, also the speed sign is an object. If we also let the truck in the above example perform some form of action, like ”braking”, the truck can be defined as an actor too: ”a car passes a speed sign when overtaking a truck that is braking”.

It is also of great relevance to clearly understand when in time different actions take place in relation to other actions. In the previously used example, ”a car passes a speed sign when overtaking a truck”, the word “when” indicates that the passing of the speed sign happened within the same time period as the overtaking activity. If “when” is instead replaced by “before”, the passing of the speed sign happens first and then the overtaking activity begins. Thus, it is important to also understand when one part of an event occur in relation to other parts of an event. Following this reasoning, different parts of an event either occur simultaneously or sequentially.

In contrast to actions (like “overtaking”) or other things that happen suddenly during an event (such as “passing a speed sign”) there are features of an event that are constant throughout the event. We call these *conditions*. For example, in an event described as “a car is overtaking a truck on a rural road on a sunny day”, both “rural road” and “sunny day” can be classified as conditions. In this case both road type and type of weather are regarded as constant during the event. This may also include many other factors describing properties of a vehicle (e.g. “a green car” or an “all-wheel drive car”), state of the driver (e.g. “driver above 50 years of age”) and so on.

The examples provided above clearly show that as more details are added to an event description, the complexity of analyzing the event, ensuring that it can be understood in an unambiguous way, increases. To enable analyzing and searching for relatively complex events, the events can be divided into smaller *event components*. This allows for searching among collected data not for a complete complex event but making separate searches for individual event components. By tagging time and place for when the individual event components occur, the event components can be re-combined into the more complex original event after event component search is completed.

Let’s have a look at one of the example events again: ”a car is overtaking a truck”. This can be divided into the event components “a car”, “overtaking” and “truck”. By doing this we don’t need to create specific search algorithms for the more complex complete event. Instead, we can focus on searching for the individual event components. We would thus need to search for “a car” an ”overtaking” activity and a “truck”, all occurring within a specific time frame and geographical space. Here we can also include searching for specific conditions that shall be fulfilled for the whole duration of the event.

However, as events get more complex, such as ”a car passes a speed sign when overtaking a truck that is braking”, we will also need a more profound understanding of what actor is actually doing what and in relation to what other actor(s) and/or object(s). Just finding a bunch of event components occurring at the same time and place is not enough. This is further elaborated in section 6.2 where the concept of Natural Language Processing is introduced.

Certain aspects differentiate these definitions from those used in EuroFOT. The definition of “conditions” (situational variables in EuroFOT) is extended to encompass all properties which can be considered constant during a certain period of time. This means that properties such as driver state and driver demographic data, as well as others, can be included. In this project, “events” are also allowed to be created without being defined from a certain number of observations from data. This allows events to be described using natural language, and all events which may be interesting can be described even if they are difficult to formally define a priori. From a natural language definition, events are not easily extracted from data, but the connection between the two is covered in the project and this is further elaborated in section 6.2.

6.1.4 Defining test events

The variance brought by the wide variety of potential actors, actions, objects and conditions that may occur in real life traffic adds complexity when defining and searching for specific events. This variance can be reduced by first studying events occurring in a less complex environment. Thus, in February of 2023 a workshop was held with all project partners with the goal of deciding how, and what kind of data should first be collected in a *controlled* environment. A test track with no other traffic, the Vårgårda airfield, was used for that purpose. A simple example is a scenario where the ego vehicle is overtaking a lead vehicle. In a controlled data collection, it will be easier to itemize data and derive initial event triggers (see section 6.1.6) that are later applied and iterated on while searching through larger naturalistic data sets. Annotation of real-life scenarios will be required at a later point, to determine the need for additional event triggers, as well as the range of the variables studied.

Controlled event development

Large scale naturalistic data, such as the data gathered in the CLOUDIA project, can be used in different ways to learn more about driving. The collected data needed to serve different objectives and therefore multiple kinds of events were required. Objectives included validation of the data collection process, data for preprocessing development, simple event scenarios for proof of concept, and more complex events that satisfy the project goals. To fulfil the objectives, events were identified using different focus areas to exemplify different end usages of the data.

Five test events were defined (see below), and then performed and recorded on the airfield. The intent was to iteratively define more events when search methods for the initial events had been developed. Data collection was conducted at the airfield with a Volvo XC90, provided by Magna, equipped with the Pionate data collection platform. The driving was performed by professional test drivers from Magna, overseen by CLOUDIA representatives from Qualcomm and Magna.

Identifying risky behavior

It is a known fact that aggressive and risky behavior contribute to a higher accident risk (Elander, West & French, 1993). The same group, approximately 15% of the population, is involved in 50% of the accidents (MacDonald, 1994; Williams, 1964). From an insurance perspective the insurance fee is based on fixed factors such as where you live, previous incidents, age, power, ADAS availability, and how expensive the car is to repair. Aggressive driving (driver behavior) is not included as a factor in today's insurance offer. With a platform such as CLOUDIA such additional factors could be explored to calculate and identify risky behavior. This could for instance include distance to other vehicles, number of sharp turns, accelerations, AEB system activations etc. A basic harsh braking event was chosen, to start off the analysis of acceleration signals and allow for further investigations of near crashes.

Example event 1: Harsh braking
Two variants where ego vehicle speed is varied at 10, 20, 40 and 70 km/h.
1a: Ego car doing a quick but planned speed reduction, no ABS activation.
1b: Ego car "panic brakes" to stop as soon as possible.

In 1a, the test drivers were instructed to make a hard but planned braking, e.g., where a driver might perceive a danger in time but still need to quickly stop the vehicle. No ABS should engage, and the situation could resemble a traffic jam.

In event 1b the drivers were instructed to "panic brake" and stop as soon as possible, which could for example correspond with animal appearing on the road. Both these events were performed multiple times with varying vehicle speed before the braking event. During the tests no other vehicles or objects were present at the runway.

Learning about crash causation and correlations

There is a lot of knowledge on accident risks in traffic, but there remains a need for continuing to learn how to prevent such accidents. Large scale data accessible via platforms such as CLOUDIA are beneficial. For instance, turning left is one of the leading causes of car crashes. 22% of all car crashes involve a left-turning vehicle and 53% of all cross-path crashes are the result of left turns (NHTSA, 2010). For bicyclists, fatal accidents are most common on country roads (Kullgren et al., 2019). When analyzing left turns, it is important to distinguish between the road turning left and there being an intersection. Map data is not always complete, thus a prototypical left turn in an intersection was included. To understand any effect on the risks posed to vulnerable road users, the more complex bicycle overtaking and car overtaking scenarios were also included.

Example event 2: Brake + turn
2a: Speed reduction followed by a 90-degree turn.
2b: Speed reduction before and during a 90-degree turn.

In version 2a the ego speed is first reduced with braking and followed by a 90-degree turn. In the second version, 2b, the braking is continued throughout the turn. The ego speed of the vehicle varied between the tests and the test driver were instructed to drive comfortably.

Example event 3: Car overtaking
Two cars are driving, one after the other, the car behind passes the other in a "flying" overtaking. Speeds for ego and target vehicle of 70 and 45 km/h and 40 and 20 km/h respectively.

For this event the target vehicle first accelerated to its target speed followed by an acceleration of the ego vehicle to a higher speed. The ego vehicle is then closing into the target vehicle and is then overtaking it, by first changing lane passing the target car and then change back to the previous lane.

Example event 4: Bicycle overtaking
A stationary bicycle is passed by a car as if driving on a rural road. Speeds of 20, 40 and 70 km/h.

For this event a stationary bicycle was mounted onto the airfield. The test drivers were instructed to do a safe overtaking as they would in naturalistic driving. The overtaking was performed at constant speed and while passing the bicycle the lateral distance to the bike was slightly increased.

Verification and clarification of effectiveness of ADAS

Different studies have shown that the effectiveness of ADAS may vary. To verify under what circumstances the effect varies, large scale data are important. For instance, it has been seen that the effectiveness of AEB for pedestrians potentially is different at different speeds and during different light conditions (Kullgren et al., 2023). Access to large scale data would be beneficial to identify what factors are associated with the lower effectiveness. This includes data on how the driver acted pre/during the event (breaking/steering) as well as current light conditions, weather, time, etc. Another example includes ACC. A major potential of ACC use is that traffic flow may become more stable and the number of shock waves which may lead to rear-end collisions can be reduced (e.g. Ciuffo et al., 2021; van Arem et al., 1996). To investigate the effectiveness of ACC and the possible reduction of near-collision-events, data such as that from CLOUDIA are necessary. Here a car following event was included to allow for further investigation of driver gaze and behavior in naturalistic driving.

Example event 5: Car following, target braking
The ego vehicle follows the target in 40 km/h when the target vehicle makes a soft speed reduction. Performed both with and without ACC active.

For this event the ego vehicle and target vehicle traveled in the same direction. When ACC was not engaged the test driver decided on a comfortable time head way. The leading car was instructed to make a smooth deceleration, which the ACC system and test driver then needed to act upon.

The test route is a 46-kilometer circuit north of Gothenburg starting off in a northwest direction then going south on E45 back to the starting point. After the test drive it was noted in a table who was driving, date, if any sunglasses were worn, and if something out of the ordinary happened in the traffic environment while driving. Additional data collection was performed on the same route in the opposite direction by a number of different Qualcomm employees. This enables comparisons of the same route with multiple drivers in the same vehicle, as well as the same route with the same driver. The vehicle was also used for additional, longer, drives by Qualcomm employees to incorporate other routes.

Naturalistic data collection

To collect additional data to the CLOUDIA project, a test vehicle and a pre-defined driving route was selected. The test vehicle was a Mercedes Benz E 220 d 4MATIC, model year 2017, which was equipped with the Pionate logging solution, Veoneer/Magna infrared DMS (driver monitoring system) and a Veoneer/Magna MVS4G front looking camera.

6.1.5 Data preparation

To allow for high quality analysis from collected data, it is essential to validate that the data quality is sufficient for the analysis intended. When reviewing the data, we found offsets in the acceleration data, both in the longitudinal and lateral directions. Over longer periods of time the average acceleration should be close to zero, since otherwise it would indicate that the velocity keeps increasing/decreasing, or the vehicle keeps turning throughout the drive. This offset also seems to change slightly between drives or even during drives, which might indicate a movement of the measuring devices. This is not unexpected given the non-invasive nature of the data acquisition platform, however, needs to be handled in pre-processing.

To reduce the noise level, a Butterworth filter of the second order with a cut-off frequency at 1Hz was used to filter all relevant signals. This reduces all high frequency components in the signal while keeping the low frequency components. To deal with the suspected calibration errors, we applied a rotation to the observed signals to compensate for the measuring device mounting. Compared to trying to remove the offsets for each signal separately, this approach has the advantage of preserving the orthogonality of the coordinate system.

6.1.6 Event trigger development

The goal of this work package is to develop a search method for complex events by breaking them up into smaller event components. The approach to perform this is to detect individual event components by searching for triggers. A trigger is a point in time that indicates that something of interest in the data occurred. The event search method can then be developed by finding out what combination or sequence of triggers are related to a specific event. But to do this we first need to specify how triggers can be detected.

Triggers can be extracted in many ways, one of the simplest ways of doing it is by looking at a signal, for example lateral acceleration, and then determine when it exceeds some threshold value. This is, however, somewhat limiting in what kind of triggers you can detect. Sometimes you may not be interested in the value at a specific time, you might want to detect a slight elevation of a signal over an extended period, or the change in a signal. This also poses limits such that it is only possible to find triggers that relates to one signal at a time. Instead, we propose to detect triggers by using cross-correlation. The cross-correlation between two signals measures how their similarity depends on their relative shift. Cross-correlation can generally be used for pattern recognition tasks, where a longer signal is searched for the occurrence of a shorter one: the kernel. This can be compared to why convolutions are used in convolutional neural networks, where features are extracted from signals based on their similarity to kernels that are learned from data. We, instead, want to design the kernels, and then detect the triggers based on the cross-correlation being higher than a threshold.

This has advantages over the simple threshold method since we are no longer limited to point-like detections. Instead, signals can be efficiently searched for features, and kernels can be designed freely. Furthermore, there is no limitation to using only one signal at a time, but it is possible to construct multidimensional kernels that combine multiple signals.

Kernel design

Even though manually designing the kernels provides the advantages of more control and better understanding of the trigger extraction, it requires more work and thoughtfulness. In this section, we describe kernels used in the present project and provide some insight in what kind of triggers each kernel is detecting. We will start with some simple kernels and work our way to the more intricate ones.

Constant kernel

The first kernel tested is a simple constant kernel extending for two seconds. This kernel detects when a signal, in average, is above some threshold for two seconds which is equivalent to performing a point like detection when the signal has been filtered with a rolling mean filter, with a window size of two seconds.

Applying this kernel to the acceleration would give us a signal that is associated with a velocity change during a short duration. This behavior is shown in Figure 3(a) where the filtered signal (green line) is very strong as the change in velocity is significant over a period. Note that this is a better measure than acceleration (blue line) itself. Although they look similar, the filtered signal with a constant kernel (green line) has significantly less noise. This is a good method to get rid of high frequency noise that is often found in raw data such as acceleration.

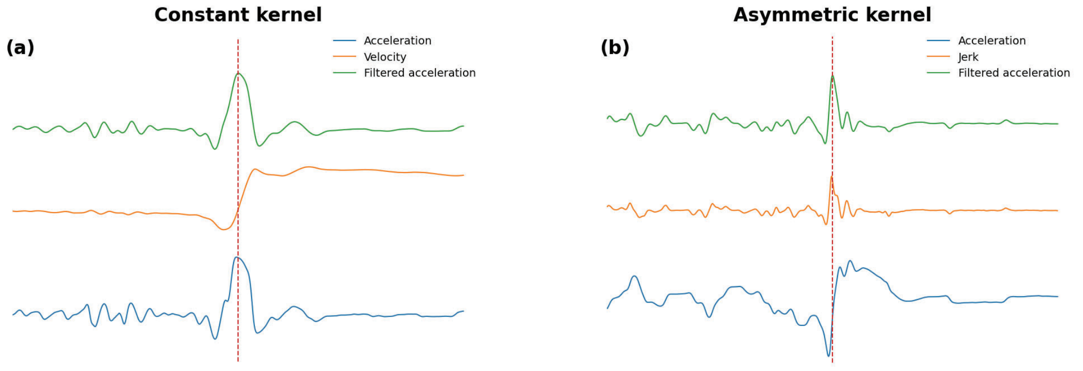

Figure 3: Application of constant and asymmetric kernels to the acceleration signal. (a) Filtered acceleration with a constant kernel (green line) is strong when the change in velocity is significant during a short period. (b) Filtered acceleration with an asymmetric kernel (green line) is strong when the change in acceleration is significant during a short period.

Applying this kernel to both longitudinal and lateral acceleration, both positive and negative, a wide range of signal behaviors can be detected. Some of the detected triggers have a lower acceleration extended for a longer time, while some have a higher acceleration for a shorter time. Positive and negative lateral acceleration triggers behave similarly, both in average behavior example triggers and their speed histogram. For longitudinal acceleration some differences are observed, the main difference being more acceleration triggers than deceleration for low velocities, and the opposite for high velocities.

Asymmetric kernel

Asymmetric kernels can be used to detect the average signal change above a threshold. Applying this kernel to the acceleration would give us a signal that is associated with a change in acceleration during a short duration. This behavior is shown in Figure 3(b) where the filtered signal (green line) is very strong as the change in acceleration is significant over a period. Note that this is a better measure than jerk (orange line) itself. Although they seem similar, the filtered signal with an asymmetric kernel (green line) has significantly less noise.

Multidimensional kernels

To allow detecting triggers in more than one dimension at the time, a more advanced multi-dimensional kernel is introduced. A multidimensional kernel can be constructed in many ways, one way is to combine the one-dimensional kernels previously presented.

With this method we can construct many different multi-dimensional kernels. We will present the detected trigger for two of the defined multidimensional kernels in more detail; Constant-Constant and Constant-Asymmetric.

In Figure 4 the Constant-Constant kernel is presented. This kernel should detect triggers with both lateral and longitudinal acceleration which is the case. From the speed histogram these triggers are predominantly detected at lower speeds.

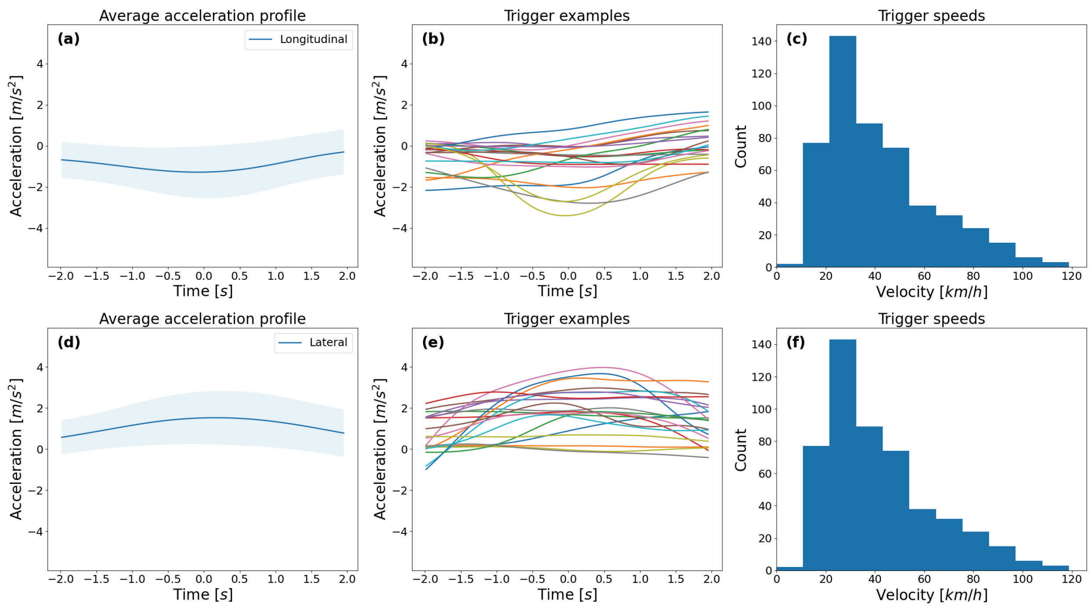

Figure 4: Average acceleration profile, example profiles and velocity histogram of detected triggers with a multi-dimensional kernel, with a constant negative kernel in longitudinal dimension and constant positive

kernel in the lateral dimension. (a)-(b) shows the acceleration profiles in longitudinal direction, (d)-(e) in the lateral direction and (c) and (f) is the velocity histogram.

The result from the next kernel, the Constant-Asymmetric kernel is presented in Figure 5. This kernel expresses a trigger indicating that a positive longitudinal acceleration and a decrease in lateral acceleration occur simultaneously. As for the previous trigger, there is a variation in the individual detections, while the acceleration behavior around the trigger, on average, is in line with what the kernel should detect. This trigger is also predominantly detected at lower speeds.

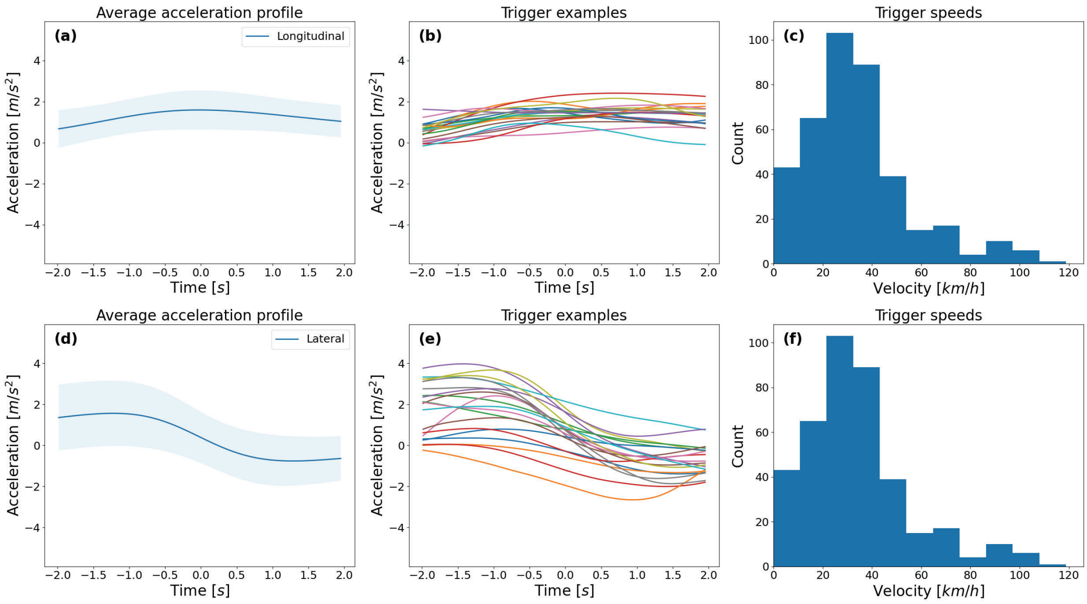

Figure 5: Average acceleration profile, example profiles and velocity histogram of detected triggers with a multi-dimensional kernel, with a constant negative kernel in longitudinal dimension and an asymmetric in lateral dimension. (a)-(b) shows the acceleration profiles in longitudinal direction, (d)-(e) in the lateral direction and (c) and (f) is the velocity histogram.

6.1.7 Triggers in example events

To investigate how the trigger formalism can be used for detecting events we study the example event defined in section 6.1.4, Controlled event development. The simplest of the example events was the harsh braking event. In Figure 6 the results of the data collection for this event are presented together with the detected triggers. The event was performed at four different speeds: 10, 20, 40, and 70 kph, and for both harsh (no ABS) and panic (with ABS) braking. The figures show that for each sample of the harsh braking events three triggers were detected which express a decrease, negative, and increase in longitudinal acceleration respectively. It thus appears that a harsh braking event can be detected using those three triggers. Detecting a braking event solely based on the deceleration trigger is not sufficient and does not express the “urgent” or “harsh” elements of the brake. The presence of decrease as well as increase triggers indicate that braking was quickly engaged and disengaged, and thus was not a slow and extended braking event.

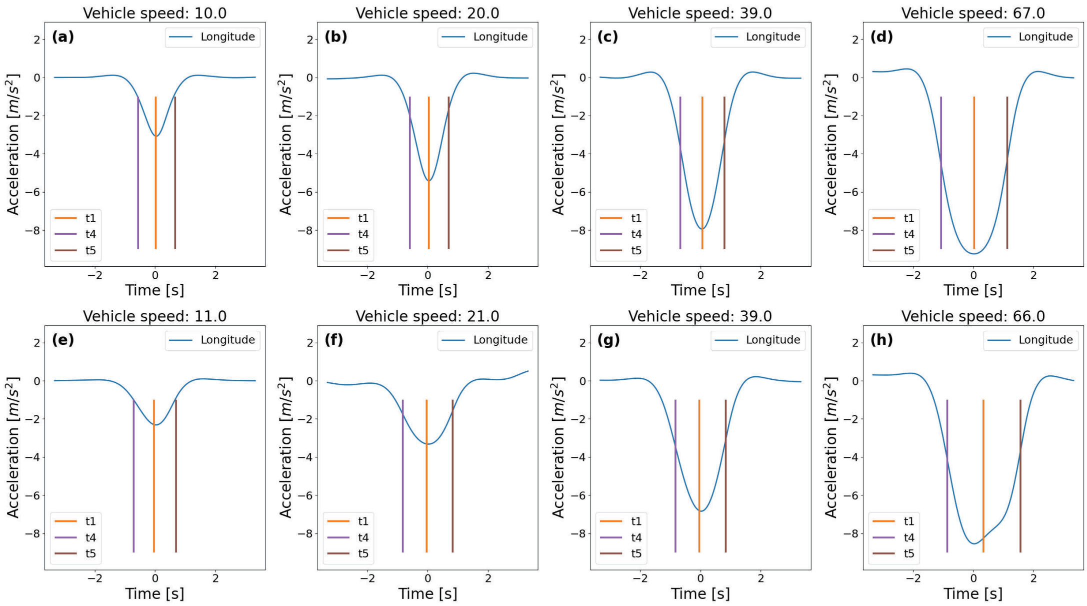

Figure 6: Acceleration and detected triggers from harsh braking events performed at the airfield. Each subfigure corresponds to one event, blue line shows longitudinal acceleration profile, and color-coded vertical lines show detected triggers. (a)-(d) show events when "panic" braking was performed (e)-(h) show harsh braking events when ABS was not activated.

Next, we study the "brake and turn" event. The acceleration and detected triggers for this event is presented in Figure 7. The event was performed four times on the airfield with different speeds, braking before or during the turn. This is a slightly more complex event, which is also seen in the number of triggers detected. All triggers are not present in all events either. It is from the figure hard to understand exactly which triggers are detected when, so a table is also presented in Table 1.

Event	Triggers ordered by time	Time difference [s]
Turn after brake: 30 kph	[t4, t1, t5, t3, t4, t1]	5.2
Turn after brake: 70 kph	[t4, t1, t5, t6, t3, t7, t4, t1]	4.2
Turn during brake: 30 kph	[t4, t7, t1, t5, t2]	1.1
Turn during brake: 70 kph	[t4, t1, t7, t5, t2, t6]	2.6

Table 1. Trigger sequences detected for annotated turn events. We also indicate the time difference between the brake trigger (t1) and the turn trigger (t2 or t3).

The table shows that the detected trigger in each event varies in both number, what triggers, and in which order they appear. However, for all occurrences of the "brake and turn" event the triggers t4, t1, and t5 are detected in that order, which is also the pattern previously associated with harsh braking. From the table it is also clear that triggers can be associated with a turn, either a lateral acceleration trigger only or a lateral acceleration trigger combined with triggers indicating an increase and decrease in lateral acceleration. The time between the triggers can then be used to distinguish between turning after braking and turning during braking.

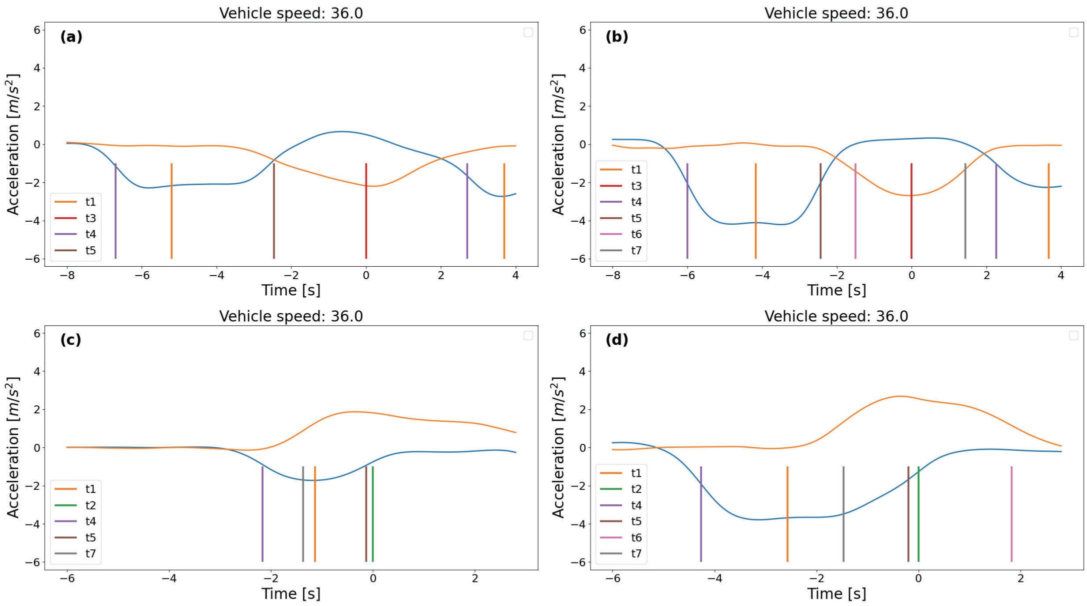

Figure 7: Acceleration and detected triggers from a brake and turn event performed at the airfield. Each subfigure corresponds to one event, blue line shows longitudinal acceleration profile and yellow lines show lateral acceleration profile, and color-coded vertical lines show detected triggers. (a)-(b) show events when a brake was followed by turn (c)-(d) show events when a brake was performed before and during turn.

The next example event is “overtaking a car”, which was performed in the airfield six times at two different ego-vehicle speeds. For this event the signal associated with an overtaking, the lateral acceleration, is too weak to be detected when using a threshold of $1m/s^2$. Therefore, in the following analysis the trigger detection uses a threshold of $0.3m/s^2$, both for the ego-vehicle and bicycle overtaking events. In Figure 8 the lateral acceleration and the detected triggers are showed for all six events.

From the figure we observe that all the occurrences of the car overtaking events have similar acceleration profiles. We see that the event seems to start and end with positive lateral acceleration and have negative acceleration in between. However, we also observed some variations both in how strong the acceleration is, especially for the different speeds, and how the negative acceleration is performed. In some cases, the negative acceleration is split into two segments, where two triggers can be detected. This indicates that the negative acceleration associated with settling into a new lane, then returning to the first lane, and is not continuous.

This variation also leads to some differences in what triggers are detected in each occurrence of the event. In all cases the event starts with the trigger sequence of t7, t2, t6 and t3, which in order expresses an increasing, positive, decreasing, and negative lateral acceleration. The continuation of the sequence varies but in general the t7, and t2 triggers are observed. The variations include presence of an extra t3 trigger, which is when the negative acceleration is split into two triggers, the presence of t6 and t7 triggers at the end of the sequence which is related to how fast the acceleration changes around last positive acceleration indicated with at t2 trigger. We also have one event occurrence where the second t2 trigger is not observed, but from the figure there seems to be some positive acceleration observed at the end of the event. This acceleration is very low, below the threshold of the trigger detection, and thus no trigger is detected. This illustrates how small the acceleration associated with an overtaking is, especially at low speeds. For event performed at higher speeds, the signal is stronger.

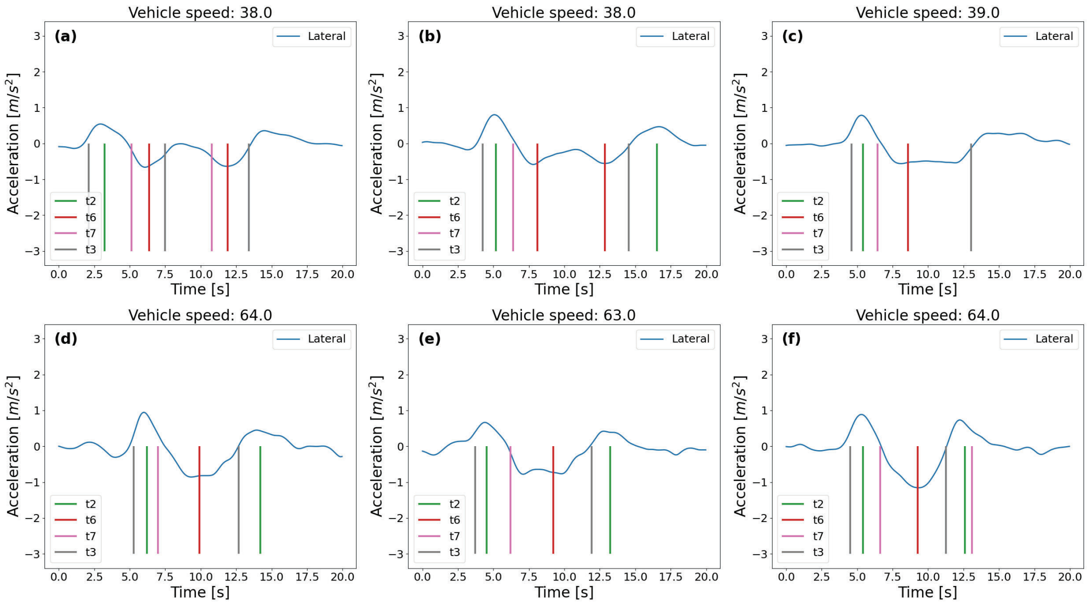

Figure 8: Acceleration and detected triggers from car overtaking event performed at airfield. Each subfigure corresponds to one event, blue line shows lateral acceleration profile, and color-coded vertical lines show detected triggers. (a)-(c) show overtaking at approximately 40 km/h, (d)-(f) at 70 km/h. Triggers detected with a threshold of $0.3\ m/s^2$.

The fourth example event is “overtaking of a bicycle”. Nine of these events were performed on the airfield for three different vehicle speeds. In Figure 9, the acceleration for all these events is presented.

The figure indicates that the acceleration profiles for bicycle overtaking is very similar to that of a car overtaking, which is expected. The events are essentially the same, a car first changes lane, passes an object and then returns to the original lane. This results in positive followed by negative and again positive lateral acceleration. Although some differences may be observed, one being that the bicycle overtaking event generally takes less time at the same speed, and two, that the negative acceleration is only observed as one peak. These differences probably occur since the bicycle on the airfield was stationary while the car overtaking was moving, making the overtaking of the bicycle take shorter time.

We can determine that the detected triggers are very similar to the car overtaking event. We also notice that very few triggers were detected in the 20 km/h events. This is due to the signals being quite small, which is especially noticeable at lower speeds. Just as for car overtaking, we generally observe the same sequence of triggers: t7, t2, t6, t3, t7, t2 and in one case a t6 trigger.

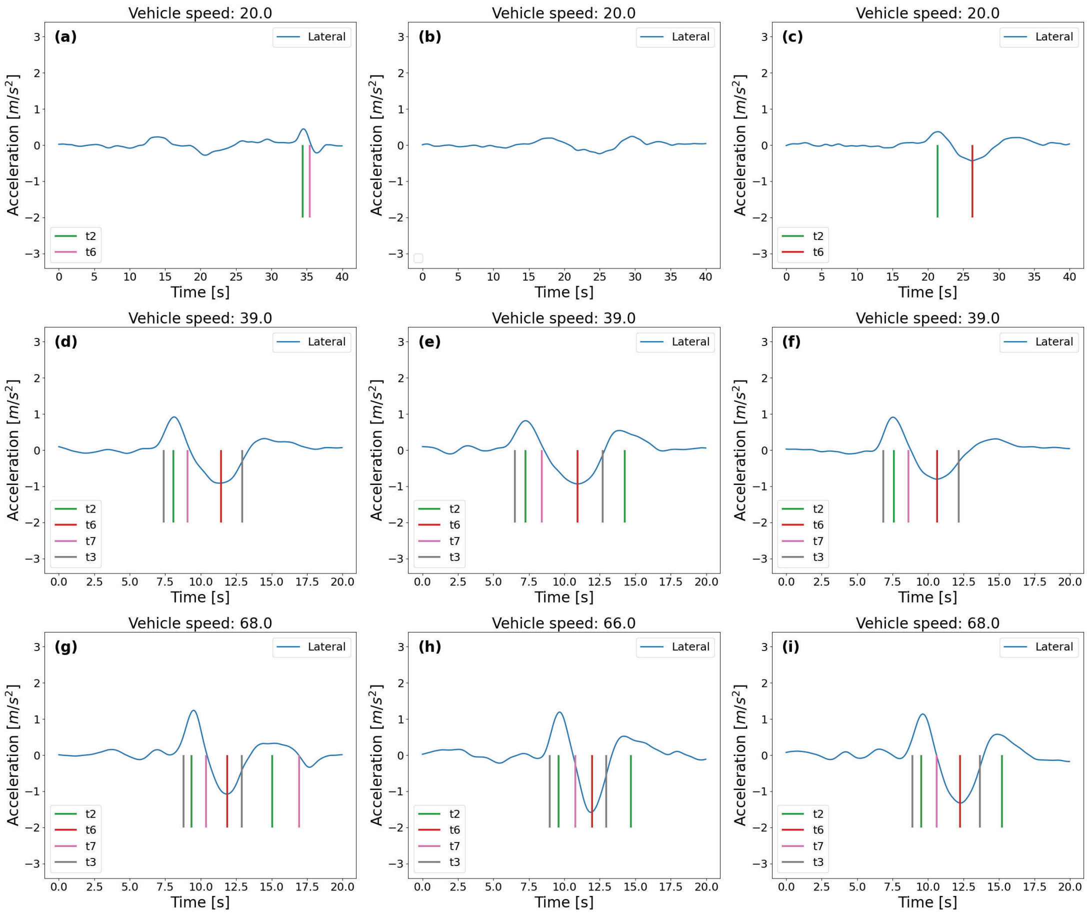

Figure 9: Acceleration and detected triggers from bicycle overtaking event performed at airfield. Each subfigure corresponds to one event, blue line shows lateral acceleration profile, and color-coded vertical lines show detected triggers. (a)-(c) show overtaking at approximately 20 km/h, (d)-(f) at 40 km/h and (g)-(i) at 70 km/h. Triggers detected with a threshold of $0.3\ m/s^2$.

The final example event is the car following situation where the target vehicle makes a sudden brake. This data from this event has unfortunately not been studied in detail, as no data from the MVS4G was available in time. The lack of object data makes this event essentially identical to the "harsh braking" event for the present analysis, and thus not prioritized.

6.1.8 Trigger-based event search

From the previous section we gained an understanding of the basic components of the car overtaking event performed in airfield. With this knowledge we want to construct a method based on the triggers to extract similar events from naturalistic driving. So, in the first iteration of the event search we look for the trigger sequence t7, t2, t6, t3, t7, t2. We exclude the last t6 trigger since it related to how fast the car returns to the original lane. We also don't search for the sequence where two t3 triggers are detected since it mostly occurred at slow speed, and most overtaking is performed at higher velocity.

To perform the event search, we start with finding all the triggers with a threshold value of 0.3 m/s^2. Then the detected triggers are filtered so only the one we are interested in are left, e.g., the one present in the trigger sequence, in this case t2, t3, t6, and t7. We then detect where the sequence of triggers matches the pattern we are interested in. If all triggers are within a time window of 10 seconds, we classify it as an overtaking event.

6.1.9 Event search in naturalistic driving data

In this section we perform the method of event search for a car overtaking event, which was described in the previous section, and evaluate the results.

The search was performed on a dataset consisting of 50 different drives, all above one hour, with a total drive time of approximately 72 hours. Since the drives are long, they are likely to be performed, at least partially, on either highways or rural roads where a car overtaking is likely to occur. When the search for the specific trigger sequence was performed, 262 events were detected. When the time constraint (10 seconds) was added, this number was reduced to 53, which would amount to 0.73 detected events per hour. The number of overtakings performed ought to be dependent on the driver's driving style and thus highly individual. With this in mind, 0.73 overtakings per hour is not obviously unreasonable.

To evaluate whether the events detected in the dataset are overtaking events or not, the driving data related to the events were investigated. As examples of some contexts observed, Figure 10, Figure 11, Figure 12, and Figure 13 are presented, each showing one instance of an event detection. The figures show lateral and longitudinal acceleration, velocity, and the car's position on a map, which together can provide a first understanding of whether the event is an overtaking or not.

In Figure 10 we observe what is likely to be an overtaking event. The lateral acceleration looks very similar to the acceleration profile observed in the example events, a rapid increase in velocity, and the location in satellite imagery appears to be a rural road. This indicates that the event is an overtaking event. However, satellite data only is not sufficient to make certain that this is a car overtaking.

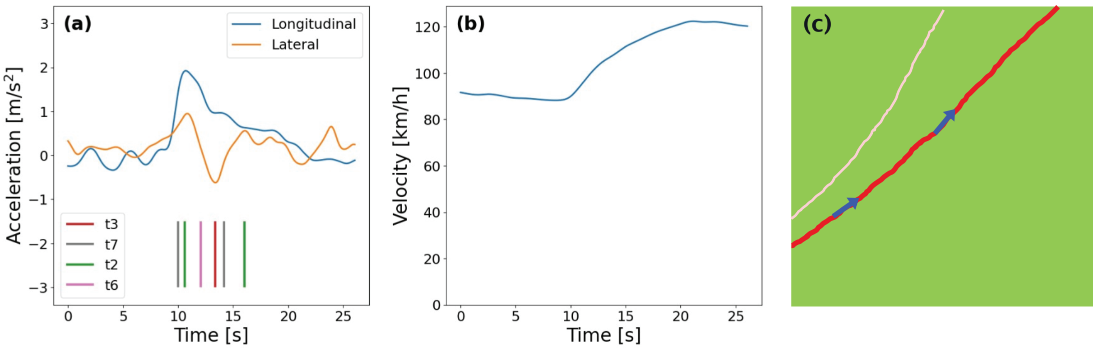

Figure 10: Example overtaking event detected using a specific sequence of triggers. (a) Longitudinal (blue line) and lateral (orange line) acceleration profile of an automatically detected event, which consists of a series of triggers (t7, t2, t6, t3, t7, t2). (b) Velocity profile of the same event. (c) Trajectory of the event plotted on a map image (red line). The direction of the travel is shown every 5 seconds (blue arrows). The arrow body (without head) represents the travel distance in 1 second. It is visible that the driver performs a

lane change and significantly increases the velocity subsequently. The lateral shift due to the lane change is also visible in the maps image overlay.

Traffic is complex and several different cases can trigger the same sequence of acceleration triggers as the overtaking event. In Figure 11 the road has a curvature which closely resembles the path of an overtaking. In Figure 12 the search method detects a lane change before a left turn in an intersection, illustrating how similar the acceleration profiles are for these events. Figure 13 may show a car overtaking, but it could also be a result of road geometry. In addition to the detected events presented, we frequently observe detections of the car going through a roundabout, which appears to have a similar acceleration profile to an overtaking. It is thus important to extend the search method with more data signals, for example object information along with the acceleration data. This will allow for more robust detection of events with fewer false positives.

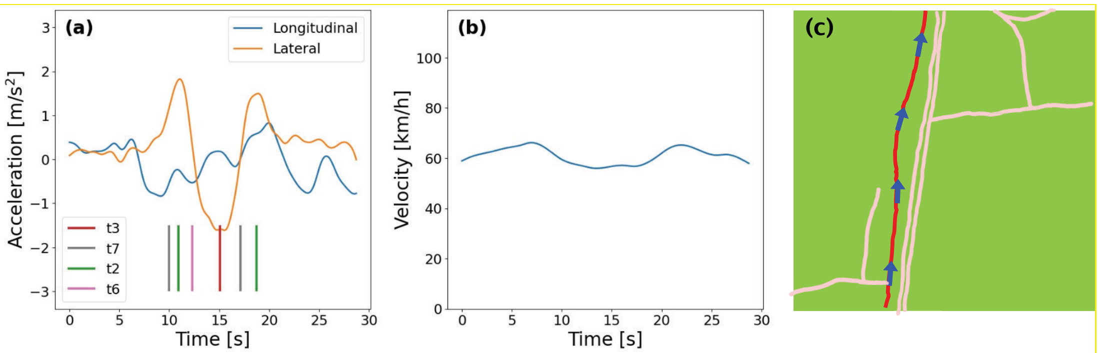

Figure 11: Detected sequence of triggers due to a road shape creating an illusion of overtaking. (a) Longitudinal (blue line) and lateral (orange line) acceleration profile of an automatically detected event, which consists of a series of triggers (t7, t2, t6, t3, t7, t2). (b) Velocity profile of the same event. (c) Trajectory of the event plotted on a map image (red line). The direction of the travel is shown every 5 seconds (blue arrows). The arrow body (without head) represents the travel distance in 1 second.

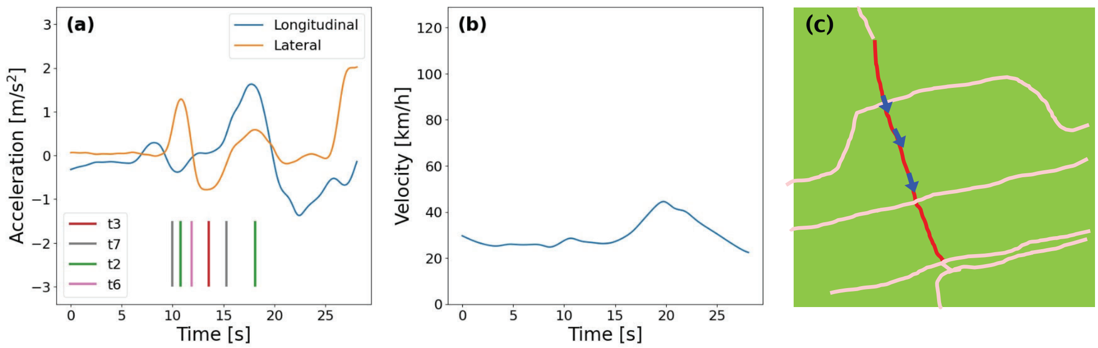

Figure 12: Detected sequence of triggers due to a lane change. (a) Longitudinal (blue line) and lateral (orange line) acceleration profile of an automatically detected event, which consists of a series of triggers (t7, t2, t6, t3, t7, t2). (b) Velocity profile of the same event. (c) Trajectory of the event plotted on a map image (red line). The direction of the travel is shown every 5 seconds (blue arrows). The arrow body (without head) represents the travel distance in 1 second.

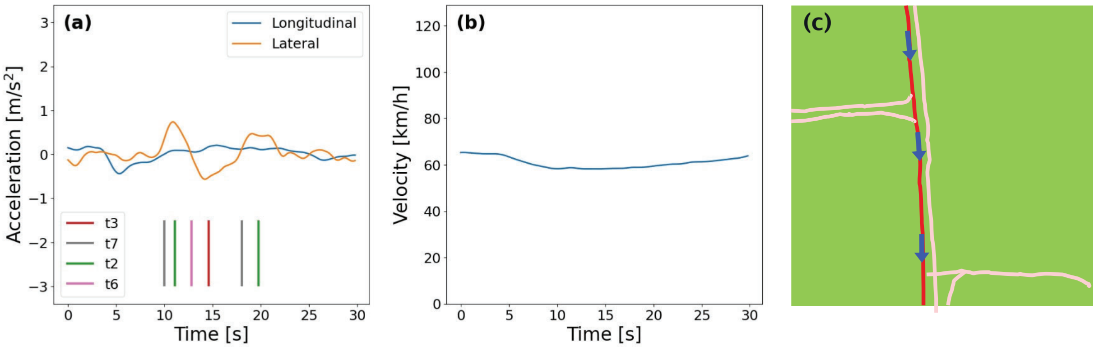

Figure 13: Detected sequence of triggers due to an unknown reason. (a) Longitudinal (blue line) and lateral (orange line) acceleration profile of an automatically detected event, which consists of a series of triggers (t7, t2, t6, t3, t7, t2). (b) Velocity profile of the same event. (c) Trajectory of the event plotted on a map image (red line). The direction of the travel is shown every 5 seconds (blue arrows). The arrow body (without head) represents the travel distance in 1 second.

6.2 Search function for specific events

With the event method at hand and (unstructured) driving data available, a search application is needed for finding the specific events of interest. In this section, different aspects of structuring and searching for data is described in more detail, including some important learnings. The section is structured in accordance with Figure 14 below. To start with, the frontend including the user interface and different options for searching for events is explained. Then, the backend including the structuring of the database is described. In the last part, some special features are described, and some aspects of scalability are covered.

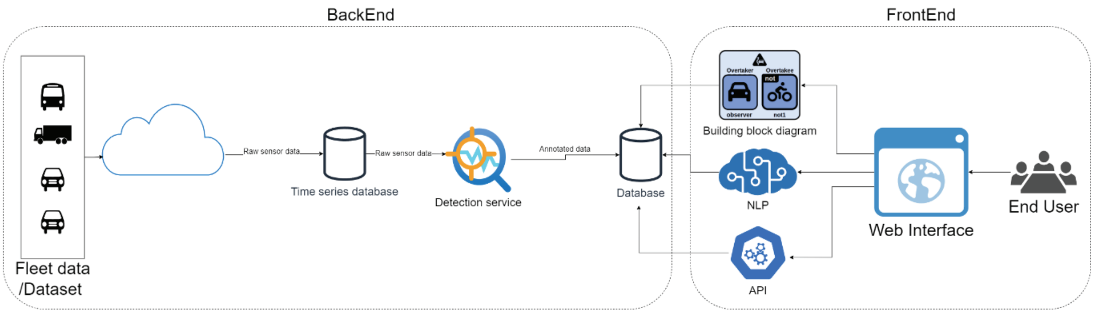

Figure 14. Backend and frontend of the search application (source: Pionate).

6.2.1 Frontend and user interface

To make searching for events easy and user friendly, a web interface is provided where users can select between three different ways of searching for specific events: Plain text (NLP), Building block diagram, or Application Programming Interface (API).

For users preferring to express an event as plain text, a simple search window is provided where an event can be described in the user's own words, see Figure 15.

The event to be searched for can be written in the search window as one or several sentences. For example, an event can be expressed as "Ego car made a left turn slowly when a bicycle was passing the road”. In order to make the content of the text searchable in the database, the system uses natural language processing (NLP) techniques. The system will then dissect the text describing the event to identify its different event components.

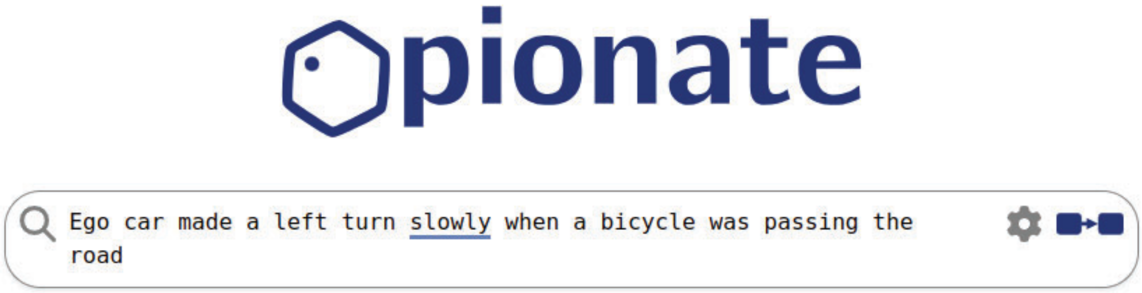

Figure 15. The plain text search window.

The first step in analyzing the text is to decompose it into its event components by identifying *time links*, which establish the temporal relationships between the different elements of an event. In this instance, the time link is represented by the word *"when"* which connects the first element (“Ego car made a left turn slowly”) to the second element (“a bicycle was passing the road”). As mentioned earlier, time links can be classified into two primary categories:

- Sequential: Elements in an event occur before or after another.
- Simultaneous: Elements of an event occur concurrently.

In the given example, the time link “when” implies a simultaneous occurrence, indicating that both elements of the event (the right turn and the bicycle passing) are happening concurrently.

To enable identifying the temporal links, a natural language processing model based on transformer architecture is used (specifically the “en_core_web_trf” large language model from spaCy (source: huggingface.co). This NLP model works in a way that the whole text is passed as an input to the model, and then the model provides annotations on the text as an output. The model performs essential tasks such as tokenization, pattern matching, part-of-speech (POS) tagging, and dependency parsing. See Figure 16 below, which is an example generated with *corenlp.run*.

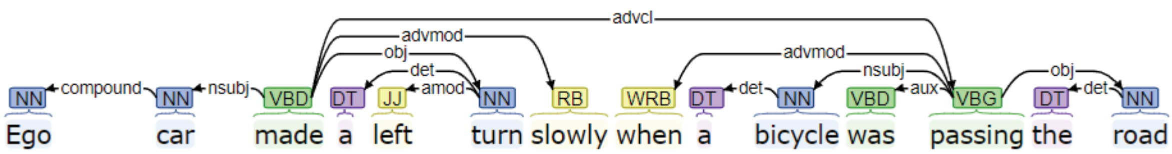

Figure 16. Example of text annotation obtained by an NLP model.

With the help of the annotations, an event text can now be decomposed into multiple event components. The model also outputs the role of each word in the text, and how words relate to each other. This can now be combined with rule based methods to break down each event element even further to obtain single event components such as individual actors, actions, objects and conditions that can be searched for in the database.

When adjectives or expressions like “at low speed” are included, the system will offer the user to specify in more detail how to delimit the search. The underlying NLP is enhanced to detect loosely defined terms like “slowly” which will make the system ask the user for further clarification. In the example illustrated in Figure 17, the user will be offered to specify the speed interval for “slowly”.

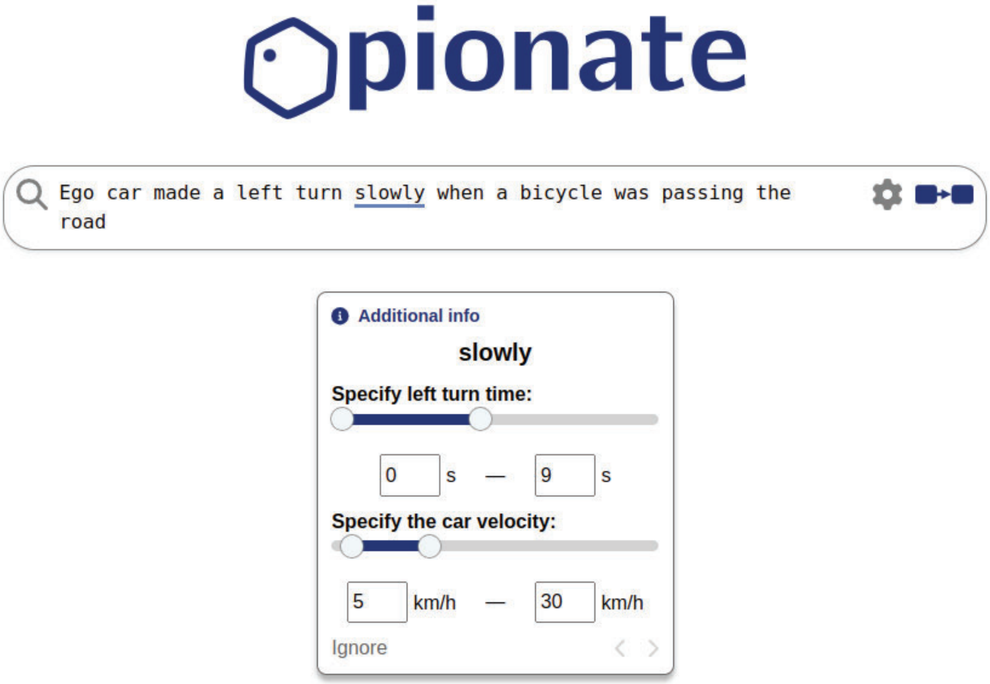

Figure 17. The user is prompted to specify some words in more detail.

Before we continue with how search results are presented, let’s briefly look into the other two alternatives for the user to describe an event to be searched for. Instead of writing plain text, there is a possibility to describe the event to be searched for using a graphical tool, see Figure 18.

The graphical tool allows users to drag and drop different icons for actors, actions, objects and conditions onto a workspace and linking them together, making it easy to build and visualize events. Each icon has properties that can be specified, like the color of a car or the direction of a turn (left or right). Once the elements are placed, users can create time links to show how event components relate to each other chronologically. Users can also adjust time markers to control when actions occur and modify variables like speed or angle for specific event components. This approach may be faster and more intuitive than describing the event in text. In the project, the graphical tool was mainly developed as a proof of concept solution for evaluation as a user interface. It has not yet been implemented as a working search tool in the system.

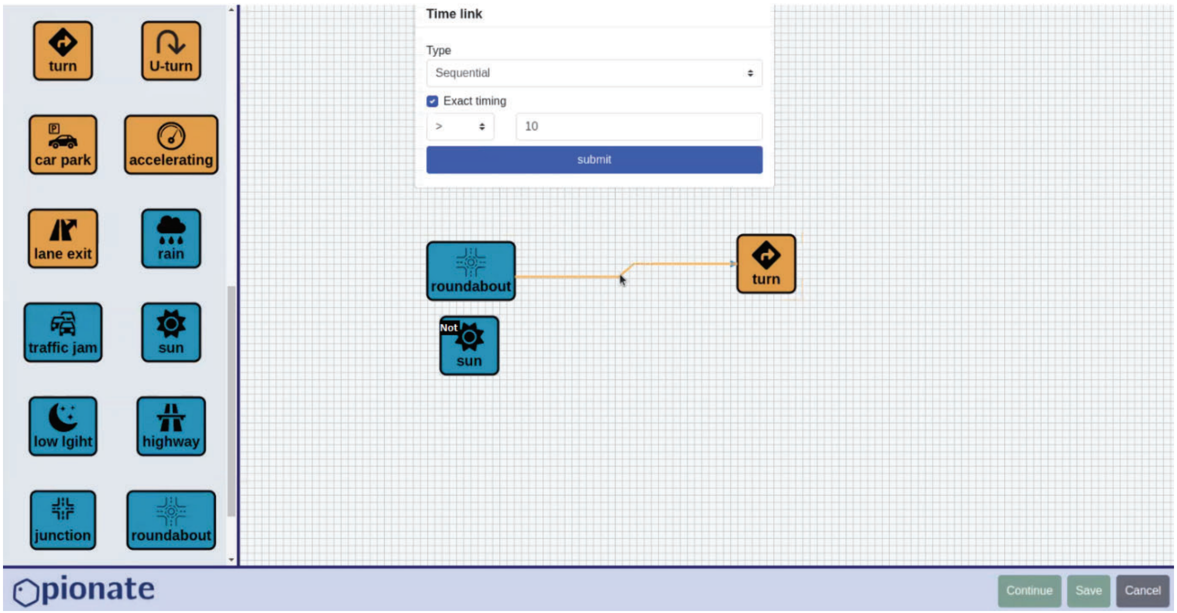

Figure 18. The graphical tool user interface for describing an event

For more advanced users familiar with coding, the system also exposes APIs that can be accessed via RESTful API calls, implemented in Python. These APIs allow users to interact with the system programmatically, bypassing the need for the drag-and-drop interface or text-based descriptions. Using the API, users can directly set up actors, actions, objects, conditions and time links with code, providing greater flexibility and efficiency when building complex scenarios. While this approach may be more complex, it offers a powerful alternative for those comfortable with programming, giving them more control and automation.

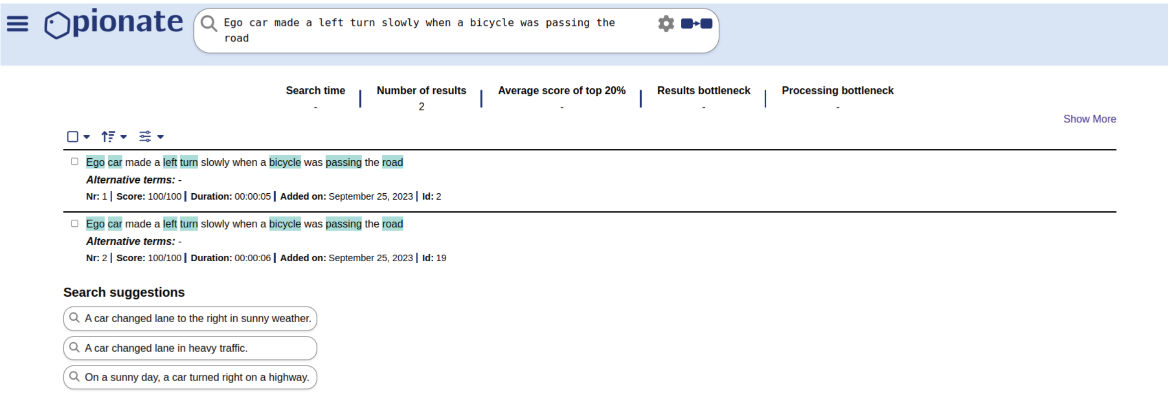

Figure 19. The search results page showing two found events.

Upon completing a search, for example by typing the event text to search for and pressing Enter, the search result page is displayed (see Figure 19). In this example, two matching events were found. Certain elements are highlighted to indicate relevance, allowing users to quickly identify key information. A score indicates how well the found event matches the searched for event. However, this function has only been briefly tested and is not yet implemented.

Upon clicking on a search result, a detailed view is displayed for that specific event, see Figure 20. It is organized to show essential information in a structured format. Top left is a map view to illustrating the event. In the example, it can clearly be seen that the car made a turn. Next, under Recording info some key details of the data origin are presented, including the Recording ID, drive duration, the recording date, the date the data was added, the recording location, the model of the ego vehicle, available sensor data, the dataset source (e.g., from a specific dataset or fleet data), and the applicable license.

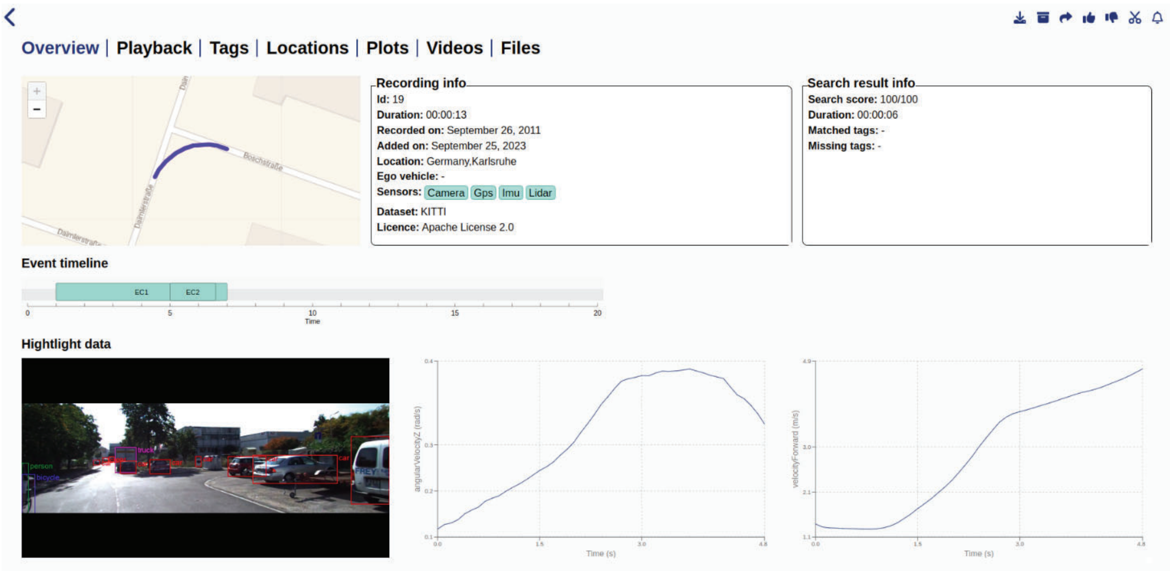

Figure 20. Detailed view for a specific event.

On the right, under Search result info, the search score is found, and the duration of the recording is displayed, followed by details on matching and missing tags, which are available if the data has been tagged previously by the user. The duration provided here describes the duration of the found event, in contrast to Recording info that shows duration for the whole recorded drive. The Event timeline illustrates the different identified event components on a timeline.

At the bottom left, camera data is displayed, enhanced with an object detection filter that highlights objects detected during the event. Finally, on the bottom row, sensor data is plotted, with the X-axis representing time and the Y-axis illustrating angular velocity around the Z-axis (showing turns) and forward velocity (relative to the car's reference system). The Y-axis metric is customizable to display any chosen sensor data.

At the top right of the detailed event result page, a row of icons are arranged, providing options to download, save, share, and give feedback (thumbs up and thumbs down button) to refine the search results. These actions are also accessible directly from the search result page facilitating efficient data management. The feedback system, which supports search refinement, is discussed in more detail in section 6.2.3.

An important concept that needs to be mentioned within this framework is the notion "ego" vehicle. The term "ego" is a property linked to an object to indicate that it refers to the vehicle that was recording the data (i.e. the car executing the right turn in the above example). This differentiation is essential for distinguishing between the car recording the data and any other car that may be part of an event (e.g. a car observed by a camera in the recording vehicle). The term “ego” is commonly used to denote the vehicle recording data. It was evident also in this project that it is important to make sure the system is making the correct semantic interpretation when describing an event.

6.2.2 Backend functions – the database

Up until now we have covered the right part of the picture, the frontend of the system (see Figure 21). Now, let’s have a look at the left part, the backend, focusing on building the database and making the data searchable.

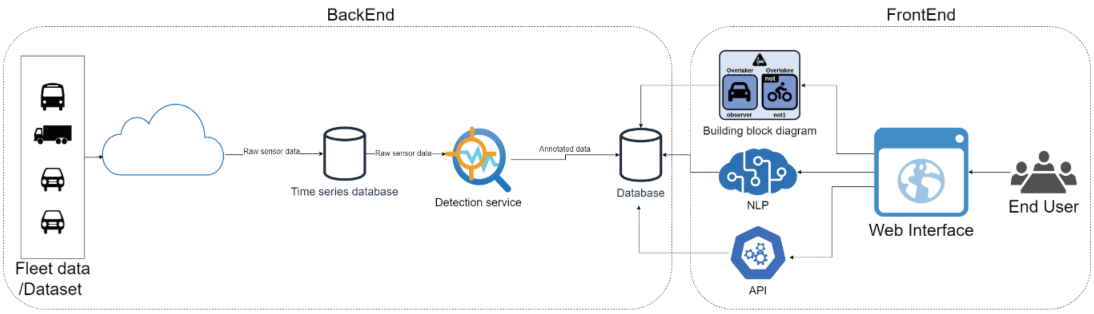

Figure 21. Backend and frontend of the search application (source: Pionate).

The database has two parts, or it can be thought of as being two distinct databases. The first is a timeseries database (on the left), optimized for storing raw data with each entry tied to a timestamp. This is where all incoming data, like recorded vehicle fleet data, is initially stored. The data can be collected from connected test vehicles or publicly available datasets (like KITTI).

Technically, this timeseries database only contains references to the actual data, indicating where it's stored on the server. Each row is a data coming from a sensor indexed by a timestamp (in milliseconds, nanoseconds, etc.) with the corresponding file path as the associated value.

In this timeseries database, a detection service runs the various detectors that filter the raw data to identify specific event components. The detectors are different sorts of algorithms that can be used to identify specific event components in the data. This is where the event triggers, as described in section 6.1.6, come into the picture. In addition to the kernel-based event triggers described earlier, detectors can also, for example, be based on machine learning algorithms providing specific image object detection capabilities. This detection process operates independently and doesn't need to be synchronized with the rest of the system. Some of the detectors used in the detection service in the project include:

- Object detection: Machine learning used for annotating camera data.
- Turn detector: A combination of several data sources including IMU and GPS for detecting that the ego vehicle is making a turn
- Brake detector: Use of IMU data for detecting braking of the ego vehicle

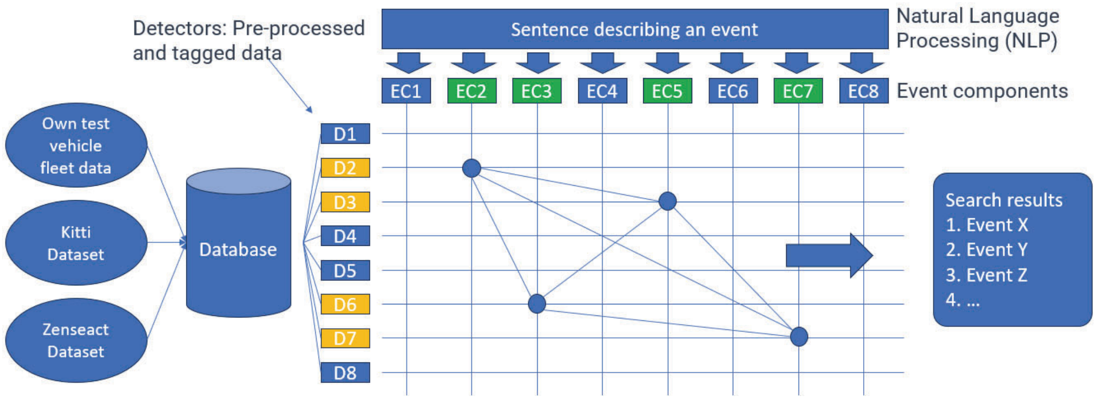

Figure 22. The combining of detectors and event components to search for events.

In Figure 22 the complete search process using detectors find specific event components is illustrated. Unstructured driving data from vehicle fleets or other datasets is fed to the database. The data is tagged by the different detectors that are used for analyzing the data. When a search text string is entered, it is analyzed using NLP to identify the specific event components. By matching event components against the tagged data (illustrated by the dots in the matrix), specific events can be detected. The time links between event components are illustrated by the lines between the dots.

A strength of this approach is that detectors can be added to the system as the system grows. User defined detectors can also be added to the system. Detectors can also be shared between colleagues or researchers working on specific events. The overall performance of the search function is always dependent on how good are the detectors that have been implemented.

As the detection service processes data through various detectors, it inserts the annotated (or cleaned) data into a separate database designed to serve the user interface (as illustrated in Figure 21). Unlike the timeseries database, this database is more structured, with each table storing key information such as the event component identifier, start time, end time, and details of the event component itself.

6.2.3 Search features and scalability

A useful feature of the search function is the "like" and "dislike" buttons (thumbs up and thumbs down) mentioned above, which lets the user provide feedback to the system, enhancing the user interface experience. This feedback system functions as a recommender, allowing users to refine their search results. For example, if a user likes a specific result, pressing the thumbs up button will lead to more similar results to be shown. Pressing the thumbs down button will lead to similar results being filtered out. In this way, the user can easily narrow the search results to those events more closely matching their preferences.

The underlying mechanism enabling this is dynamic time warping (DTW), a technique used to compare results regardless of their duration. DTW analyses the shape of the data and performs pattern matching, allowing the system to identify similarities between two timeseries data even when they have different durations or if they have different offsets. Figure 23 below shows how DTW differs from simply performing Euclidian matching.

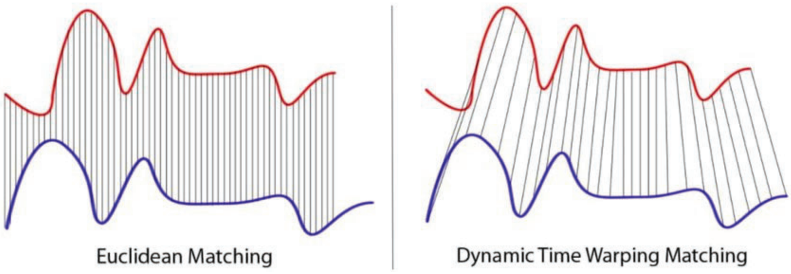

Figure 23. Illustration of Dynamic Time Warping (Source:researchgate.net)

In this way the users can refine the search results by liking or disliking events they are interested in rather than alternating the textual event description.

The database is designed for scalability, operating across multiple machines to efficiently handle growing storage demands. The system supports large datasets and can incorporate both public and proprietary data. Scalability has been tested with the KITTI and ZOD datasets, along with two internal collections: one from an airfield and another from in-house recorded data, totaling 73 drives. This confirms the system's capability to manage and scale effectively with diverse data sources.

Scalability testing has been conducted to simulate conditions equivalent to handling 6.2 terabytes of data, representing approximately 330 hours of driving. Under these conditions, the system has demonstrated its ability to manage large data loads while maintaining performance. In a worst-case scenario, a search operation would take up to 15 seconds using the current approach.

6.3 Driving data collection platform and test vehicles

The data collection platform played an important role for supplying the project with driving data. In addition to collecting naturalistic data from driving on public roads, it also enabled data collection from specific pre-defined events that were performed on a test track. The data collection platform can be installed in a plurality of car brands without necessitating a collaboration with the vehicle manufacturer for data access. In the project, the data collection platform was installed in three different test vehicles (a Volvo XC90, a Mercedes E220d, and a BMW 330e, see Figure 25).

The data collection platform used in the project (see Figure 24) is an upgraded version of the platform developed in the projects Mini Fridge (Diarienr. 2019-05212) and Imaginate (Diarienr. 2020-02793). While the core hardware remains the same, a new type of camera sensor was added, the software for collecting and uploading the data to the server was upgraded, and a new casing was developed that made the system more robust and helped to simplify the installations.

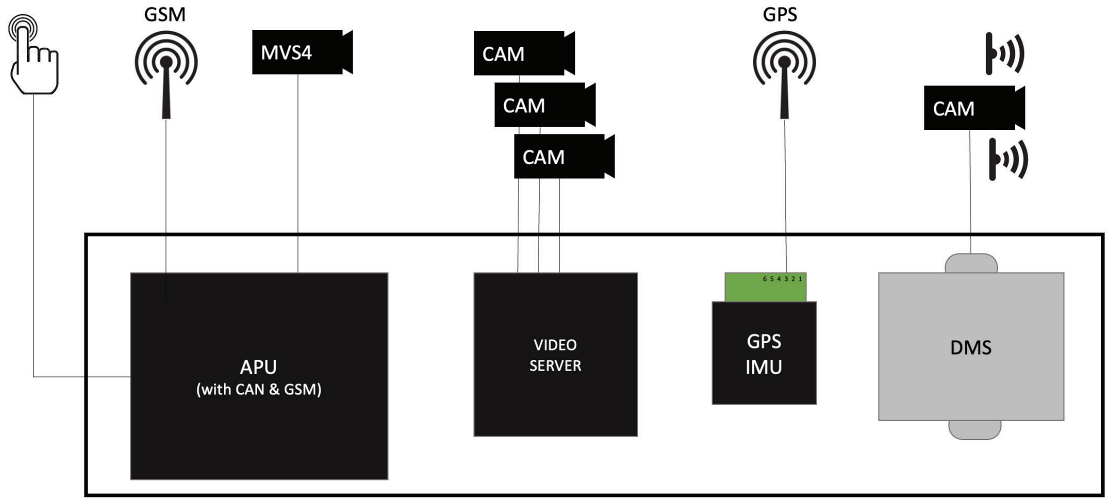

Figure 24. The data collection platform that was installed in the test vehicles.

The heart of the platform is an accelerated processing unit (APU), where a central processing unit (CPU) has been combined with a graphics processing unit (GPU) on the same chip. In this way, the APU provides relatively high performance in a compact format. The APU has been equipped with a 1TB solid state drive (SSD) that is used for the operating system as well as onboard storage of data. It was also equipped with a 4TB SSD for additional onboard data storage capacity. The APU was also equipped with ethernet ports, additional cards for CAN communication and a 4G modem.

In order to keep track of the test vehicle’s geographical position and movements the platform is equipped with a programmable sensor module for positioning (GNSS) and orientation determination. It has a satellite receiver, a magnetic field sensor, an accelerometer, and a gyroscope. The orientation is measured as accelerations in x, y, z directions as well as yaw, pitch, and roll. The data is sent to the APU via CAN at 10Hz update rate.

Up to four video cameras can be connected to the logging platform. In this project three cameras with 1080p resolution were used. One camera was mounted on the sun visor and directed towards the driver, one was placed inside the windscreen filming in front of the car, and one filmed the instrument cluster. By filming the instrument cluster important information about the vehicle status (e.g. ACC tuned on) can be retrieved by using image recognition algorithms. It does not require any specific connection with the vehicle. Fisheye lenses on the cameras enables a wide field of view but such lenses also somewhat distort the images. The cameras are connected to a dedicated video server that in turn streams the video data to the APU via an ethernet connection.

As a complement, a fourth camera from Magna (MVS4G) was included, looking in front of the vehicle. This camera is equipped with special software enabling real time object detection information in addition to the video stream. The object information is sent to the APU via CAN. To make the object detection software in this camera work properly it needs to be fed with speed information from the vehicle's CAN network. However, in this case the speed information could be obtained from the logging platform's GNSS instead. Thus, no need for connecting to CAN in the car.

To track the driver's gaze behavior, a driver monitoring system (DMS) was included in the logging platform. The DMS uses IR diodes, and a camera placed on the dashboard to track the driver's eye movements. In this way driver attention and behavior (e.g. driver looking at the cell phone) can be studied in more detail. The eye tracking information is sent to the APU via CAN.

Included in the platform is also a push button that can be placed close to the driver. By pushing the button, the driver can choose to immediately stop all data logging by the platform. However, as long as the test vehicle is still running, the data collection platform will continue to process and upload already logged data.

The logging platform cabinet was placed in the trunk of the test vehicle and all that was needed to make it run is power from the vehicle (12V DC). Main part of the installation job is the mounting of sensors (cameras) and routing of cables between sensors and the data logging platform. Also, antennas for GNSS and 4G need to be installed.

Figure 25. Magna test vehicles at the airfield in Vårgårda.

As soon as the platform is powered, i.e. when the test vehicle is started, the APU boots and connects to the server via the 4G modem. It also starts collecting data from the different sensors that are connected to it. All collected data is time-stamped with the global GPS time, enabling efficient post processing as well as post-synchronization of data collected from other sources, such as other test vehicles. All collected data is stored on board on the SSDs as a backup. Simultaneously, collected data is also uploaded via the 4G link directly to the server. Under favorable conditions, approximately 2GB of data can be uploaded per minute, but this may vary depending on the quality of the connection. When the test vehicle is turned off, a timer keeps the system running for a few more minutes to ensure that all logged data is securely saved. In addition to uploading data, the 4G connection enables continuous monitoring of the test vehicles (e.g. when they are driven and where) as well as over the air updates of the software used on the platform.

6.4 Simulation of large number of connected vehicles

The possibility to collect driving data from test vehicles has been important for conducting this project. Due to limited resources, the number of test vehicles in the project had to be limited (three in total). This fulfilled the needs for data collection in the project. However, the event database is intended to be used also with large fleets of vehicles collecting data. Therefore, simulation was used to emulate a large number of simultaneously connected vehicles.

When viewed in a broader perspective, the development of new ADAS functions require collection of data from large number of vehicles. The safe roll-out of increasingly advanced driver assistance systems at SAE Level 2 and 3 and, more importantly, to prepare the coordinated roll-out of systems aiming at SAE Level 4 and up, require the continuous monitoring of key performance indicators (KPIs) of the system's maturity on the edge. These KPIs include a growing amount of information perceived from a vehicle's surroundings as modern ADAS systems, such as AEB or LKAS, use for example cameras and radars to safely support the driver. If such systems' operational design domains (ODDs) are extended, field evidence of such a system's robustness is needed. This can only be effectively achieved at scale in a reasonable amount of time by getting data from thousands of vehicles, and hence, fleets of prototypical cars, test vehicles, and, in essence, ride hailing fleets that are meant to prepare the transition to robotaxis, would create a near future growing demand of a flexible, cloud-enabled data analytics environment to continuously calculate such KPIs.

In this project, we have systematically evaluated such a scenario by using simulations as a real vehicle fleet at that scale was not accessible. We used a cloud-enabled "scale-on-demand" environment to simulate data streams from initially ten thousand simulated vehicles. This unveiled restrictions with respect to network and data flow management at the cloud side.

Initially, this constraint could have been easily mitigated by network segmentation that would have, however, ended up with a potential single-point-of-failure if one segment would become unavailable or non-responsive. Instead, we designed and evaluated a different setup where an edge-facing gateway would handle and act as load balancer to the vehicle fleet. Any incoming connection to synchronize data from the edge platform installed in vehicles to the cloud (handled by one of three actual backend systems) for data analytics. Each of the backend nodes was capable of handling up to 341,000 incoming data streams. Thus, in this way, we were able to successfully simulate data uploads from over 1 million vehicles in a fleet. The actual tests were performed by using Amazon

Cloud where virtual servers were set up to emulate the large number of test vehicle platforms connecting with the data collection cloud server.

This setup allows for scalability in two dimensions: more edge-to-cloud backends to potentially scale to more vehicles as well as more cloud backends to spread the data storage, pre-processing, and analysis based on business optimization criteria like: (a) regulatory constraints based on geographical regions, (b) dynamically responding to flexible pricing options in the cloud environment (e.g., utilizing discounted cloud resources in case of overprovisioning from the cloud provider), and (c) separating the geographical edge-to-cloud end-point, bringing it closer to the actual operating area of the vehicles (i.e., to meet legal constraints in specific countries) but separating the data processing to either on-premise installations outside the major cloud operators or to conduct cloud analytics in regions with different prices or regulatory constraints.

6.5 Summary of the results and connection to FFI targets

The project has delivered on the objectives set out in the beginning of the project and the main project targets, described as the project cornerstones (see Figure 26), have been met.

A method for defining events, based on events, event components and detectors has been developed (1). The data collection platform has been upgraded and the scalability of the system has been tested with simulations of over one million connected vehicles (2). A search function for finding specific events has been developed, and it can be accessed via an easy-to-use interface based on graphical building blocks, free text search or an API (3). The event search platform has been demonstrated, for the project partner companies, for some other automotive companies and to researchers focusing on traffic safety (4).

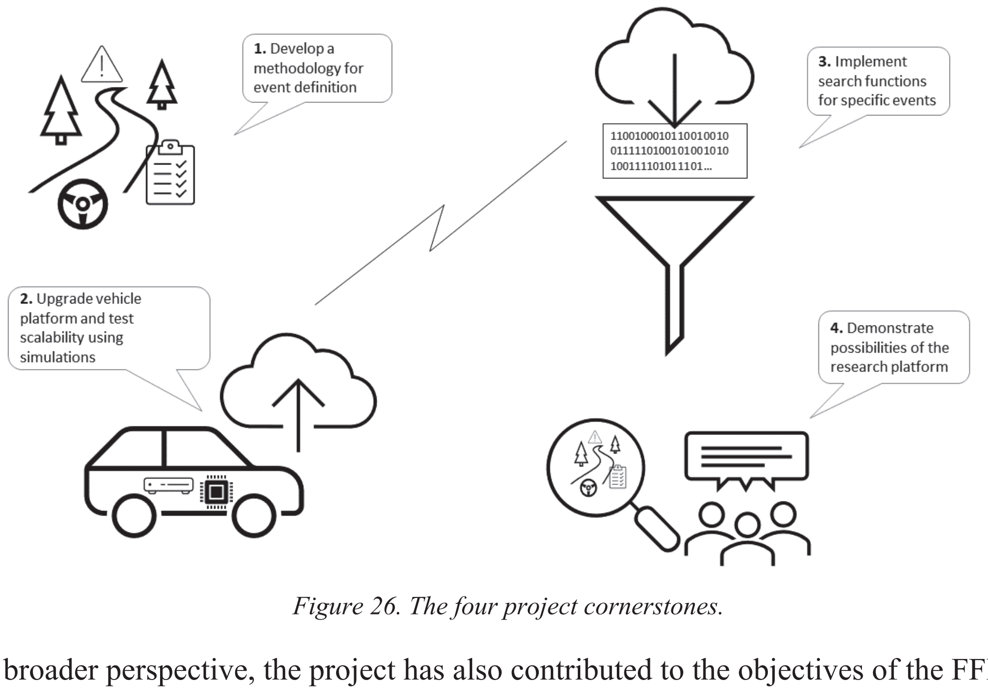

Figure 26. The four project cornerstones.

In a broader perspective, the project has also contributed to the objectives of the FFI program in several ways including:

- Developing a method and system supporting the development of new automotive safety systems and functions, which in turn can contribute to strengthening the competitiveness Swedish automotive industry.
- Connecting to international actors, especially since during the course of the project the Swedish automotive supplier Veoneer was divided into two companies leading to an involvement and collaboration with international suppliers Magna and Qualcomm.
- Supporting collaboration with small and medium sized companies, in this case involving the startup company Pionate.
- Supporting collaboration between automotive suppliers, in this case Magna, Qualcomm and Pionate.
- Supporting collaboration between different lines of business. In addition to automotive suppliers, also insurance company Folksam has been part of the project. Further, one interesting conclusion from the project is that the developed methodology and system for searching for events could also be applied in other types of business, such as finance, healthcare and authority administration.
- Involving researchers from universities, since the project members from Pionate have strong connections to Chalmers and Gothenburg University and the project member from Folksam is connected to ITRL at KTH. This has also enabled some research results emanating from academical work to be incorporated in this project.

The project has also delivered on several of the focus areas outlined for the Safe Autonomous Driving FFI sub-program, including for example:

- Verification and development techniques for the development of safe automated functions, such as ADAS that has been in focus in this project.
- Monitoring and understanding driver behavior, driver perception and the use of different safety-related systems.
- Human machine interaction, both in terms of how drivers perceive the driver environment and ease of use for the event search system developed in the project.
- Methods and tools for representation of driver behavior, where the event search system enables detailed analyses of driver behavior in specific driving situations.
- Fast and efficient communication between the vehicle and infrastructure, enabling near real-time data from connected vehicles.
- Methods and technologies connecting vehicles with cloud services where machine learning and other techniques can be used for developing safer automated driving.

7 Dissemination and publications

How are the project results planned to be used and disseminated?	Mark with X	Comment
Increase knowledge in the field	X	Knowledge has been increased regarding how to define events, how to efficiently collect data from test vehicles and how to search for specific events in large amounts of data.
Be passed on to other advanced technological development projects	(X)	It is possible that some more advanced engineering will be done, but more likely a product development project closer to market adaptation.
Be passed on to product development projects	X	Pionate will aim for continuing to develop the database and server functionality, preferably in close collaboration with one or more potential customers.
Introduced on the market	(X)	Eventually, the goal is market introduction, but first some more product development is needed.
Used in investigations / regulatory / licensing / political decisions		

The project results have been disseminated in the form of demonstrations/seminars arranged for other members of the project partner organizations, some other automotive companies as well as for researchers in the field of traffic safety. This also provided valuable feedback for the project. In January 2024 the project was presented at the Transportforum conference in Linköping. Pionate have also produced a series of short videos demonstrating the event search function.

A master thesis has been performed in connection with the project where naturalistic driving data was studied. The title is "Evaluating Car Crash Risk Changes with pay-as-you-speed Insurance - an insurance service case study in Sweden". The author's name is Peiling Wu, from Aalto University School of engineering and KTH Royal Institute of Technology. The thesis was initiated by project partner Folksam and focused on evaluating data from an earlier field study of 600 participants where the impact on speeding and crashes from a system monitoring vehicle speeding was studied.

8 Conclusions and future research

The CLOUDIA project has focused on developing methods for defining and efficiently searching for specific traffic events in large amounts of unstructured naturalistic driving data. In brief, the project has developed a method for defining *events*. By splitting an event into *event components* based on the categories actors, activities, objects and conditions, event components can be made searchable in the database. To search for event components, specific algorithms are used, referred to as *detectors*. A user interface for searching for events has been developed. Further, the scalability of the system has been evaluated. Here we list some important conclusions from the project.

- The project has shown that events can be efficiently searched for, saving time for users that need to find specific driving data. The search function can also be developed to allow users to subscribe for specific events, so they immediately get a notice when a specific event has occurred in a test vehicle in the field.
- By applying a modular approach, i.e. breaking complex events down into event components, it is much easier to develop the detectors. Conversely, with an increasing number of detectors, the number of potential events that can be composed increases exponentially. Also, by combining more detectors, the searches can be more confident thus reducing the number of false positives and false negatives.
- The project has demonstrated how detectors can be defined as kernel-based triggers, A kernel can be designed to express very versatile behavior in the data signals. By intricate design, both local information and information over longer timescales in timeseries data can be expressed. Using cross-correlation, kernels can be used on any data signals like acceleration, velocity, gaze behavior and object information, on continuous as well as discrete signals. Also, the use of machine learning for object detection in video streams has been applied.
- While the search function lets each user (company) develop their own detectors, there are possibilities to share detectors between users. In this way, not all users need to develop all detectors on their own, and users can compare detector performance across different datasets.
- The project has demonstrated the scalability of the system developed for searching for events. This includes scalability in terms of number of test vehicles connected to the system. Simulations using external servers showed that the system could handle more than one million vehicles simultaneously. Scalability has also been demonstrated in the sense that data from other datasets (like KITTI and ZOD) could be added to the database and be made searchable with the same search tools.
- It was important to the project to collect data from own test vehicles. This enabled the project to define very specific test events for controlled environments, in addition to other publicly available datasets.
- A simple and easy-to-use user interface makes searching for events fast and efficient. By applying Natural Language Processing methods (NLP), events can be described as

plain text by the user. NLP will then decompose the event into searchable event components. As an alternative, a user interface using graphical building blocks was tested. Also, an application programming interface (API) can be offered for users that prefer coding.

- The term “ego” proved to be important to make the system interpret an event description correctly, clearly distinguishing between the vehicle collecting the data (the “ego vehicle”) and any other vehicle in an event.
- By including functions for further delimiting the search results, using dynamic time warping techniques (DTW), the user can easily tell the system what type of search results are wanted and not.

In addition to these conclusions, the project has generated many ideas on what further research can be performed in this area.

- As the detectors are key for what events can be searched for, detector development is always important. One specific area, among several, that can be expanded on is to develop detectors for analyzing human behavior based on driver models.
- While this project has been much focused on automotive applications and the development of ADAS, the involvement of an insurance company in the project provided interesting insights on other potential use of this type of search tools. It could be used for, for example, city planning or other types of business/authorities where large amounts of data need to be analyzed, such as medicine, human behavior data, tax offices or in the financial sector. Another potential use could be to search for data that can be used for creating different types of digital twins.
- By adding modules using generative AI, the system could be used for re-creating similar but slightly modified events. In this way, using real event data as a basis, different “what if” events can be created and tested.
- Integrate functionality for investigating alternative search terms. Sometimes it is difficult for a user to describe an event in a way that the expected or wanted event search results are obtained. If the user starts with a “best guess” description, an AI agent can be used for proposing alternative search strings, interactively helping the user to get closer to what is really wanted. An interactive AI agent can , for example, tell the user that “do you know that your most limiting parameter right now is your definition of ‘slowly’? Can I recommend a broader interval?”
- Add functionality for indicating *accountability* for presented search results. When using deep learning techniques, accountability is difficult to obtain since results are “black box” generated. When an AI system is instead designed as a set of small AI modules, each one fully explainable, it is referred to as “glass box” AI, or “interpretable” AI. By illustrating to what extent, a search result is based on “black box” and “interpretable” AI, the user can be provided with an indication of the accountability.

9 Participating parties and contact persons

The following organizations have been involved in the project.

pionate	Pionate AB	Fredrik von Corswant, fredrik.von.corswant@pionate.com
MAGNA	Magna Electronics Sweden AB (formerly Veoneer Sweden AB)	Oliver Brunnegård, oliver.brunnegard@magna.com
Qualcomm	Qualcomm Technologies Sweden AB (formerly Veoneer Sweden AB)	Annika Larsson, alarsson@qti.qualcomm.com
Folksam	Folksam Ömsesidig Sakförsäkring	Maria Klingegård, maria.klingegard@folksam.se

10 References

Ciuffo, B., Mattas, K., Makridis, M., Albano, G., Anesiadou, A., He, Y., Josvai, S., Komnos, D., Pataki, M., Vass, S., Szalay, Z. (2021). Requiem on the positive effects of commercial adaptive cruise control on motorway traffic and recommendations for future automated driving systems. *Transportation Research Part C: Emerging Technologies*, vol 130, https://doi.org/10.1016/j.trc.2021.103305.

Elander, J., West, R., & French, D. (1993). Behavioral correlates of individual differences in road-traffic crash risk: An examination of methods and findings. *Psychological Bulletin*, 113(2), 279–294. https://doi.org/10.1037/0033-2909.113.2.279

Etemad, A. (coordinator). (2021). *Final Project Results*. L3Pilot Deliverable D1.7

Faber, F., E. Jonkers, M. Ljung Aust, M. Benmimoun, M. Regan, S. Jamson, J. Dobberstein. (2012) *Analysis methods for user related aspects and impact assessment on traffic safety, traffic efficiency and environment*, EuroFOT Deliverable D6.2

Huggingface.co https://huggingface.co/spacy/en_core_web_trf

Kessler, C., Etemad, A., Alessandretti, G., Heinig, K., Selpi, Brouwer, A., Cserpinszky, A., Hagleitner, M. B., and Benmimoun, M. (2012). *Final Report*. EuroFOT Deliverable D11.3

Kullgren A, Stigson H, Ydenius A, Axelsson A, Engström E, Rizzi M. (2019). The potential of vehicle and road infrastructure interventions in fatal bicyclist accidents on Swedish roads - What can in-depth studies tell us? Traffic Inj Prev. 2019;20(sup1):S7-S12. doi: 10.1080/15389588.2019.1610171. PMID: 31381450.

Kullgren, A., Khabat, A., Tingvall, C., (2023) Effects on crash risk of automatic emergency braking systems for pedestrians and bicyclists. Paper nr: NO.23-0114-O. 27thESV2023. Yokohoma, Japan.

Larsson, A. & Ryman, M., (2023). CLOUDIA WP1 report, internal report, Arriver Software AB, a Qualcomm Technology Inc. subsidiary.

MacDonald, W. (1994). Young driver research program: A review of information on young driver crashes. Report CR 128. Canberra, Australia: Federal Office of Road Safety

Metz, B., Rösener, C., Louw, T., Aittoniemi, E., Bjorvatn., A., Wörle, J., Weber, H., Antas Torrao, G., Silla, A., Innamaa, S., Fahrenkrog, F., Heum, P., Pedersen, K., Merat, N., Nordhoff, S., Beuster, A., Dotzauer, M., Streubel, T. (2019). *Evaluation Methods.* L3Pilot Deliverable D3.3

NHTSA (2010). *Crash Factors in Intersection-Related Crashes: An On-Scene Perspective.* DOT HS 811 366. https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/811366 Accessed on 9 October 2023.

Pegasus. (2019). *PEGASUS method – an overview*. Technical Report. Accessible through https://www.pegasusprojekt.de/files/tmpl/Pegasus-Abschlussveranstaltung/PEGASUS-Gesamtmethode.pdf Accessed on 3 October 2023.

Researchgate.net https://www.researchgate.net/figure/Simple-visualization-of-dynamic-time-warping-DTW-alignment-Instead-of-assuming-a_fig2_350397933

Van Arem, B., Hogema, J. H., Vanderschuren, M. J. W. A., & Verheul, C. H. (1996). *An assessment of the impact of autonomous intelligent cruise control.* TNO-report INRO-VVG 1995-17a

Veoneer and SwissRe, 2021. The significance of progress – Implications of the diversity and complexity of ADAS for road safety and the insurance landscape. White Paper.

Weber, H., Hiller, J., Eckstein, L., Metz, B., Landau, A., Mun Lee, Y., Louw, T., Madigan, R., Merat, N., Lehtonen, E., Sintonen, H., Innamaa, S., Streubel, T., Pipkorn, L., Svanberg, E., van Weperen, M., Hogema, J., Bolovinou, A., Rigos, A., Junghans, M., Zhang, M., Pastor Trullos, J., Zerbe, A., Schindhelm, R., Page, Y., Hagleitner, W., Zlocki, A. (2021). *Pilot Evaluation Results.* L3Pilot Deliverable D7.3

Williams, N. (1964). Traffic accidents – Epidemiology and medical aspects of prevention. *Canadian Medical Association Journal.* vol 90, p1099 – 1104.