

Interdisciplinary Project

# Accident Detection on the A9 Test Stretch Using Roadside Sensors

Unfallerkennung auf der A9 Teststrecke mittels sich  
am Straßenrand befindenden Sensoren

<b>Supervisor</b>	Prof. Dr.-Ing. habil. Alois C. Knoll
<b>Advisor</b>	Walter Zimmer, M.Sc.
<b>Author</b>	Marc Pavel
<b>Date</b>	March 15, 2024 in Munich



# Disclaimer

I confirm that this interdisciplinary project is my own work and I have documented all sources and material used.

Munich, March 15, 2024

---

(Marc Pavel)

## **Abstract**

The leading cause of death for people aged 5-29 years is road traffic injuries resulting in approximately 1.19 million total deaths each year [23]. Mitigating human errors such as speeding, driving under the influence, and distracted driving as well as reducing the delay in detecting accidents and providing care for the involved participants is inevitable to reduce the number of road traffic fatalities [23]. Since modern vehicles are equipped with Light Detecting and Ranging (LiDAR) and Radio Detection and Ranging (radar) sensors, accident detection systems have gained considerable popularity. However, this view is often limited due to other vehicles or objects blocking the Field of View (FOV). Therefore, this paper proposes firstly an event log consisting of anomalous traffic situations recorded with roadside sensors, secondly a scenario mining approach, which detects maneuvers in the recorded data focusing on finding accidents with the help of a rule-based approach and thirdly an automated accident detection process to mine large amount of traffic data and extract important features. The automated accident detection was executed on the rosbags recorded in 5 months, where a total of 831,969 unique vehicles, 3,748 standing vehicles in a driving lane, 138 standing shoulder vehicles, 120 breakdowns, and 1 accident were detected. On a test dataset from the event log, the rule-based accident detection achieves a precision of 100% and a recall of 33% due to its limitation of only detecting rear-end collisions, which can be extended in the future. Additionally, one new accident was found in the recorded data of five months due to the automated accident detection.

## **Zusammenfassung**

Die führende Todesursache für Menschen im Alter von 5-29 Jahren sind Verkehrsunfälle, die jedes Jahr zu etwa 1,19 Millionen Todesfällen führen [23]. Die Minderung menschlicher Fehler wie erhöhte Geschwindigkeit, Fahren unter dem Einfluss von Alkohol oder Ablenkung sowie die Verringerung der Verzögerung bei der Unfallerkennung und der Bereitstellung von Hilfe für die Beteiligten ist unvermeidlich, um die Anzahl der Verkehrstoten zu reduzieren [23]. Da moderne Fahrzeuge häufig mit LiDAR- und Radarsensoren ausgestattet sind, haben Unfallerkennungssysteme erhebliche Popularität erlangt in den letzten Jahren. Jedoch ist das Sichtfeld der Kameras oft von anderen Fahrzeugen oder Objekten begrenzt. Daher schlägt diese Arbeit erstens ein Ereignisprotokoll vor, das aus anomalen Verkehrssituationen besteht, die mit Straßensensoren aufgezeichnet wurden. Zweitens wird ein Szenario-Mining-Ansatz vorgestellt, der Manöver in den aufgezeichneten Daten erkennt und sich darauf konzentriert, Unfälle mithilfe eines regelbasierten Ansatzes zu finden. Drittens wird ein automatisierter Unfallerkennungsprozess vorgestellt, um eine große Menge von Verkehrsdaten zu analysieren und wichtige Merkmale zu extrahieren. Die automatisierte Unfallerkennung wurde auf den in 5 Monaten aufgezeichneten Rosbags durchgeführt, wobei insgesamt 831.969 unterschiedliche

Fahrzeuge, 3.748 stehende Fahrzeuge auf einer Fahrspur, 138 stehende Fahrzeuge auf dem Seitenstreifen, 120 Pannen und 1 Unfall erkannt wurden. Auf dem Testdatensatz aus dem Ereignisprotokoll erreicht die regelbasierte Unfallerkennung eine Präzision von 100% und eine Rückrufquote von 33%, aufgrund ihrer Begrenzung auf die Erkennung von Auffahrunfällen, welche in der Zukunft erweitert werden kann. Zusätzlich wurde ein neuer Unfall in den aufgezeichneten Daten von fünf Monaten aufgrund der automatisierten Unfallerkennung gefunden.



# Contents

- 1 Introduction** **1**
  - 1.1 Context . . . . . 1
  - 1.2 Problem . . . . . 2
  - 1.3 Contribution . . . . . 2
  
- 2 Terms and Definitions** **5**
  - 2.1 Robot Operating System (ROS) . . . . . 5
  - 2.2 Data and Scenario Mining . . . . . 5
  - 2.3 OpenDRIVE . . . . . 5
  - 2.4 Rule-based Approach . . . . . 5
  - 2.5 Deep Learning . . . . . 6
  
- 3 Related Work** **7**
  - 3.1 Scenario Mining . . . . . 7
  - 3.2 Rule-based Systems . . . . . 7
    - 3.2.1 Rule-based Accident Prediction . . . . . 8
    - 3.2.2 Rule-based Accident Detection . . . . . 8
  - 3.3 Machine and Deep Learning-based Accident Detection . . . . . 8
  
- 4 Methodology** **11**
  - 4.1 Event Log . . . . . 11
  - 4.2 Scenario Mining . . . . . 15
    - 4.2.1 Feature Extraction . . . . . 16
      - 4.2.1.1 Feature: Lane ID . . . . . 16
      - 4.2.1.2 Feature: Distance to Leading and Following Vehicle . . . . . 17
      - 4.2.1.3 Feature: Average Velocity / Average Velocity per Frame . . . . . 18
    - 4.2.2 Maneuver Detection . . . . . 20
      - 4.2.2.1 Label: Standing Shoulder . . . . . 20
      - 4.2.2.2 Label: Traffic Jam / Slow-moving traffic . . . . . 20
      - 4.2.2.3 Label: Breakdown . . . . . 21
      - 4.2.2.4 Label: Accident . . . . . 22
    - 4.2.3 Scenario Statistics . . . . . 23
  - 4.3 Automated Accident Detection . . . . . 23
  
- 5 Evaluation** **27**
  - 5.1 Metrics . . . . . 27
  - 5.2 Structure . . . . . 29
    - 5.2.1 Image Extraction . . . . . 29
    - 5.2.2 Deep Learning-based Accident Detection . . . . . 30
  - 5.3 Results . . . . . 31
    - 5.3.1 Scenario Mining . . . . . 31

5.3.1.1	Speedup . . . . .	31
5.3.1.2	Functional Correctness . . . . .	32
5.3.2	Automated Accident Detection . . . . .	36
5.3.2.1	Precision . . . . .	38
5.3.2.2	False Negative Analysis . . . . .	40
5.3.3	Comparison with a Deep Learning-based Accident Detection . . . . .	41
5.3.3.1	Precision . . . . .	41
5.3.3.2	Recall . . . . .	41
5.3.3.3	Runtime . . . . .	42
5.4	Discussion . . . . .	42
<b>6</b>	<b>Outlook</b>	<b>45</b>
<b>7</b>	<b>Conclusion</b>	<b>47</b>
	<b>List of Abbreviations</b>	<b>49</b>
	<b>List of Figures</b>	<b>51</b>
	<b>List of Tables</b>	<b>53</b>
	<b>Bibliography</b>	<b>55</b>

# Chapter 1

## Introduction

### 1.1 Context

In this era of rapid technological advancements, road safety has gained extraordinary significance. Figure 1.1 depicts the number of road traffic fatalities in the EU from 2000 up to 2022. With cars as well as roads getting safer over the years, an immense decrease in road traffic fatalities can be supervised in the last 20 years going from 51.400 in 2001 down to 18.800 in 2020. However, EU’s target is to reduce the number of casualties even more down to 11.400 in the year 2030, since the leading cause of death for people aged 5-29 years is road traffic injuries [23].

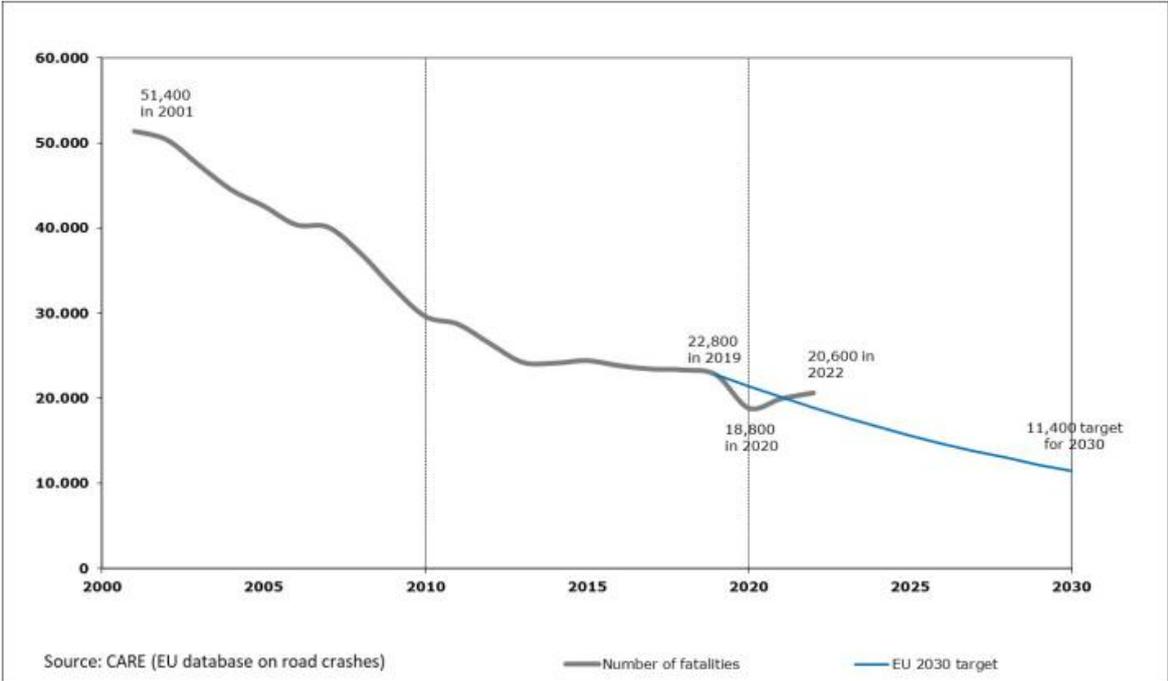


Figure 1.1: Trend in the number of road traffic fatalities in the EU [6].

The risk factors on the road comprise speeding, driving under the influence, distracted driving, etc. [23]. Accommodating these human errors is inevitable to ensure road safety. An additional risk factor, which is not related to humans, is the delay in detecting accidents and providing care for the involved participants [23]. Traditional accident reporting mechanisms such as radio or Google Maps suffer from delays and rely on humans reporting accidents

[32]. The integration of advanced technologies such as artificial intelligence and real-time sensor recordings presents a promising solution for an automated accident detection model mitigating these challenges.

## 1.2 Problem

To ensure safer traffic behavior and faster emergency response time, an accident detection model must be developed. With the help of an accident detection model, an anomaly traffic dataset can be created with which different accidents can be analyzed to create an accident prediction model extending the Advanced Driver Assistance Systems (ADAS) of autonomous vehicles. However, this requires the autonomous vehicles to perceive their environment.

Therefore, this paper is written in cooperation with ATUOtech.agil, which is a research project in the area of autonomous driving. Nowadays, modern vehicles are equipped with LiDAR and radar sensors to record the environment around the vehicle. However, this view is often limited due to other vehicles or objects blocking the FOV. This project aims to tackle this challenge by recording the environment with roadside LiDAR, radar, and camera sensors, which are placed on sign bridges above the vehicles along the highway. With this approach, a digital twin can be created in real-time mimicking the traffic on the highway by detecting and tracking traffic participants. The digital twin can then be sent and integrated into the traffic participants' sensory system to ensure safer autonomous behavior on roads by suggesting safer traffic routes early on or warning of potential accidents. [3]

To train the accident detection model for this use case, a dataset consisting of accidents on the A9 test stretch is needed. However, only 3 accidents are known in the recordings. Due to the lack of training data for the accident detection model, the need for a simple approach, that can detect potential dangerous traffic situations, arises. With the help of this approach, a larger anomaly traffic dataset can be created to train a more sophisticated accident detection and prediction model in the future. The goal is to develop an accident detection model, which can detect accidents reliably, meaning it achieves a minimum precision of 0.7 and a recall of 0.85. The recall is more important since the aim is to find as many accidents as possible. Additionally, the accident detection model should be real-time capable meaning it can analyze at least 25 Frames Per Second (FPS). Therefore, the following research question arises for the simple accident detection approach:

1. "Is it possible to reliably detect accidents on the A9 test stretch using roadside sensors?"
2. "Is it possible to detect accidents on the A9 test stretch in real-time using roadside sensors?"

## 1.3 Contribution

The focus of this work is the creation of an automated rule-based accident detection on the recorded rosbags from the A9 test stretch. The aim is to enable the creation of an anomaly traffic dataset, which enables a more thorough analysis of road anomalies, especially accidents.

This paper extends the work of Aaron Kaefer [1] and is closely related to the work of Daniel Lehmberg. It consists of the following 3 contributions:

1. **Event Log:** An event log containing all detected anomalies in the recorded rosbags from the A9 test stretch and functioning as training data for the scenario mining.
2. **Scenario Mining:** A scenario mining approach, which firstly extracts the raw recordings from the rosbags, secondly extracts features such as the lane ID of traffic participants and the distance to the leading and following vehicle and then thirdly detects important maneuvers on the recorded data. The focus lies on maneuvers, which help to detect accidents in the rosbags, such as standing vehicles on the shoulder lane, traffic jams, slow-moving traffic, and most importantly accidents themselves. Once the maneuver detection is finished, the scenario statistics can be created from the detected maneuvers to label all rosbags into different categories such as traffic jams, breakdowns, or accidents.
3. **Automated Accident Detection:** With the help of the scenario mining, a framework for an automated analysis process for the detection of accidents in an arbitrary amount of rosbags, stored in a cloud storage, is being proposed. This framework can be used to classify rosbags to create a larger anomaly traffic dataset in the future.



# Chapter 2

## Terms and Definitions

### 2.1 Robot Operating System (ROS)

The Robot Operating System (ROS) is a meta operating system consisting of a set of software libraries and tools to build robot applications [26]. The general idea of ROS is to have different ROS Nodes, which communicate with each other via ROS Topics or ROS Services to transfer data. Firstly, to use ROS, a ROS Node needs to be created, which is an executable file in a ROS package programmed in C++ or Python. ROS Nodes are then communicating with each other over ROS Topics. This means every ROS Node can subscribe to a ROS Topic to receive all data, which is being sent on the ROS Topic, and every ROS Node can publish to a ROS Topic to send all subscribing ROS Nodes data. The data sent over the ROS Topics is called a ROS Message. Multiple ROS Messages can then be stored in a single Bag. Tools like rosbag are used nowadays to create Bags, which store all serialized data published to specific ROS Topics in a file. The ROS Bags can then be played back, once the need arises.

### 2.2 Data and Scenario Mining

Scenario mining is closely correlated to data mining. The goal of data mining is to process large volumes of data with modern computing devices [21]. With the help of sophisticated automated techniques, the hidden information, which can provide deeper insights and reveal invaluable knowledge, can be extracted from these large databases [21]. On the other hand, scenario mining defines the process of finding scenarios in a scenario data set, which fulfill certain criteria e.g. an accident occurred or a standing vehicle is present in the scenario [10].

### 2.3 OpenDRIVE

OpenDRIVE is a widespread and well-established standard that defines the `xodr` file format. With the help of the Extensible Markup Language (XML) syntax, it defines a common base for describing the geometry of roads, lanes, road marks, road signs, etc. [2].

### 2.4 Rule-based Approach

A rule-based approach is a heuristic approach, which is built upon hand-constructed rules by analyzing the deviations of generic properties of a good product [7, 29]. Due to the experts'

knowledge, which was analyzed to extract the rules, they work well for the problems they are intended for [14]. Compared to other machine learning approaches or even deep learning approaches, rule-based approaches require fewer data to generate these rules making them applicable to use cases with a lack of training and test data [5].

## 2.5 Deep Learning

Deep learning is a subfield of machine learning focusing on artificial neural networks and algorithms inspired by the structure and function of the human brain. The difference to machine learning is that deep learning refers to neural networks, which have multiple layers enabling the neural networks to learn to represent data with multiple levels of abstraction and the structure in large datasets [24]. This is done by using the backpropagation algorithm to change the internal parameters specifying which features should be learned from the input. However, compared to machine learning algorithms deep learning requires a large amount of training data to achieve desirable results. Nowadays, deep learning is used in state-of-the-art speech recognition, visual object recognition, object detection, and many other domains [17].

# Chapter 3

## Related Work

Accident detection and prediction in transportation systems is a critical aspect of ensuring road safety and reducing the impact of road incidents. This chapter reviews existing literature and research on current state-of-the-art accident detection systems, shedding light on various approaches, methodologies, and technologies employed in this domain.

### 3.1 Scenario Mining

Kaefer et al. [1] proposed a scenario catalog, a scenario generation framework, a scenario extraction pipeline, a scenario augmentation framework, a maneuver detection as well as scenario mining, scenario statistics, and lastly scenario simulation and visualization. While their work did not focus on accident detection and prediction, they solved the questions of how to create an extensive collection of driving scenarios, which are relevant for the assessment of automated vehicles. Therefore, they proposed a scenario mining strategy, which is responsible for analyzing raw data recordings stored in rosbags and then applying a feature extraction. The implemented features are lane ID, offset from lane center, and distance to leading and following vehicle. With the help of the features, different maneuvers are then detected, which are lane change, cut-in and cut-out, tailgate, speeding, standing vehicle, weather, right, left, and U-turns, straight at crossing as well as enter and exit the highway. Due to the structure, the scenario mining approach is easily extendable of an accident detection and the respective necessary features and maneuvers [1].

### 3.2 Rule-based Systems

Rule-based accident detection and prediction systems leverage predefined sets of rules and conditions to identify, classify, and predict accidents or anomalous events. These systems often rely on a combination of vehicle dynamics, environmental data, and surveillance technologies to formulate rules for accurate accident detection and prediction. Nowadays, the domain of rule-based accident detection systems based on data, which monitors the traffic situation, is a popular research field [4, 12, 15].

### 3.2.1 Rule-based Accident Prediction

Ye et al. [35] proposed a solution for traffic accident prediction by considering the improvement of traffic safety efficiencies. They use the extended belief rule-based system, which is one of the most popular data-driven systems and was first introduced by Liu et al. [20]. The extended belief rule-based system is responsible for constructing the rules from input-output data pairs and calculating the rule weights for every rule based on the Euclidean distance. The advantages of the extended belief rule-based system are the ability to exploit experts' knowledge to enhance data analytics and having a visible inference traffic accident prediction process as well as interpretable prediction results. However, for the extended belief rule-based system to formulate these rules the recorded accident data from the A9 test stretch is not sufficient [35].

### 3.2.2 Rule-based Accident Detection

A real-time and rule-based traffic accident detection system, which uses wireless sensor networks, was proposed by Sherif et al. [30]. This approach uses sensors, which are installed in the vehicle, to detect accidents and the vehicles right before as well as the amount of passengers in the vehicle. Then, this information is sent to a monitoring station, which is responsible for tracking the location, where the accident occurred, and directly alerts the authorities of the potential injuries to save as many lives as possible. However, this approach requires every vehicle to have the necessary equipment of sensors present in the vehicle making it costly to implement [30].

Another approach for a rule-based accident detection system was proposed by Sheu et al. [31], which consists of three different components. Firstly, incident symptom identification, which uses knowledge-based logical rules to recognize anomalous changes in raw traffic data differing from incident-free cases. Secondly, a signal processing step is executed which uses stochastic estimation of incident-related lane traffic characteristics and thirdly, the pattern recognition for the incident detection is applied to the output of the second step. Additionally to the incident detection, this approach provides information on the blocked lanes, the start and end time of the incident, lane-changing fractions, queue lengths in blocked lanes, and the number of vehicles in each adjacent lane. However, it was specifically implemented for the use of point detectors to estimate real-time lane-changing probabilities and the change in queue-length [31].

## 3.3 Machine and Deep Learning-based Accident Detection

Recent years have witnessed a paradigm shift with the widespread adoption of machine and deep learning for accident detection [18]. Decision trees, naive Bayes, Multilayer Perceptron (MLP)s, Convolutional Neural Network (CNN)s, Recurrent Neural Network (RNN)s, etc. have demonstrated remarkable capabilities in learning intricate patterns and representations from raw sensor data [11, 16, 36]. Several studies have explored the application of machine and deep learning in accident detection using different types of input data, including images, videos, and sensor readings [22, 28, 33].

Pillai et al. [27] propose a real-time image enhancement for an automatic accident detection through the use of Closed Circuit Television (CCTV) footage using deep learning. Their approach is split up into three stages and takes images as input. Firstly, for the detection stage Mini-You Only Look Once (YOLO) is implemented, which is a deep learning model architecture with reduced model size and computational overhead compared to its big brother YOLO. It was trained using knowledge distillation and achieves comparable accuracy to YOLO. Secondly, the tracking stage uses Simple Online Real-time Tracking (SORT) to track the detected vehicles and their status of the damage variable, which is triggered once the vehicle is involved in a crash. Thirdly, in the classification stage each segmented vehicle from the images is classified based on the damage variable. Multiple machine learning algorithms were compared in this stage with a Support Vector Machine (SVM) with a radial basis kernel yielding the best performance. However, their approach is limited to the static notion of accidents by classifying damaged vehicles [27].

A similar approach which takes videos as input instead of images is proposed by Li et al. [18]. It comprises the following modules: a detection module, which uses a Faster Region-based Convolutional Neural Network (R-CNN) for object detection, a background modeling module responsible for removing driving cars from the background by using Mixture of Gaussian (MOG), a mask extraction module, which extracts the parts of the road where anomalies are likely to occur and most importantly a multi-granularity tracking, which comprises of a box level tracking branch and a pixel level tracking branch. This approach does not only focus on accidents, but detects anomalies in the road network such as vehicle breakdowns, standing vehicles, etc. [18].

Another approach, which uses videos as input to detect anomalies, is the fast and unsupervised anomaly detection in traffic videos from Doshi et al. [8] consisting of three modules. Firstly, the preprocessing module outputs all stationary objects detected in the input video. Then, the candidate selection module applies the nearest neighbor approach to remove any misclassified stationary objects and afterward makes use of K-means clustering to spot potential anomalous regions. Finally, the backtracking anomaly detection algorithm computes the structural similarity between the region of interest and the traffic participants to classify anomalies [8].

While current accident detection systems utilize state-of-the-art machine and deep learning-based accident detection systems, the amount of traffic anomaly and accident datasets are scarce [14]. Therefore, the need for an automated accident detection system arises, which only requires a few training instances and can detect accidents in the recorded data from the A9 test stretch.



# Chapter 4

## Methodology

In this section, the methodology for an automated rule-based accident detection approach on the A9 test stretch is shown in detail. It consists of the following contributions:

1. Event Log

The Event Log consists of 32 rosbags, which were labeled by hand. They cover multiple traffic scenarios such as standing vehicles on the shoulder lane, breakdowns, and accidents, which are being used to create the rules for the different maneuvers in the Scenario Mining.

2. Scenario Mining

The Scenario Mining is responsible for first extracting the raw recordings from the rosbags into data frames from which the necessary features for the use case of autonomous driving such as the lane the vehicles are driving on, the distance to the leading and following vehicle, and the overall average velocity on the highway can be calculated. Next, it considers the features to detect maneuvers, such as standing vehicles on the shoulder lane, traffic jams or slow-moving vehicles, any potential breakdowns, and most importantly accidents. Afterward, the scenario statistics can be created for every rosbag comprising all detected maneuvers. With the help of the statistics, a dataset containing anomalous traffic situations can be generated.

3. Automated Accident Detection

The Automated Accident Detection is responsible for executing the accident detection on a specified amount of recorded rosbags. The aim is to filter out all rosbags, which are not relevant for the creation of an anomaly traffic dataset. Therefore, the scenario statistics are being calculated and stored for every rosbag. Additionally, the images in which a breakdown or accident is detected are extracted from the rosbags to get a better understanding of detected anomalies.

### 4.1 Event Log

Figure 4.1 depicts the Event Log containing 14 different traffic situations recorded from the test stretch on the A9 highway. For each traffic situation the date, event, and description is

being stored to ensure easy use of the rosbags in Section 4.2. The following labels are present in the Event Log to split the rosbags into important traffic situations:

- **Emergency Vehicle:** An emergency vehicle, which is in operation, can foreshadow a potential accident.
- **Vehicle Transport:** A vehicle transporter might be detected as multiple vehicles. This can lead to False Positive (FP)s in the accident detection.
- **Tow Truck:** A tow truck might be detected as multiple vehicles if it is towing another vehicle. This can lead to FPs in the accident detection.
- **Shoulder:** The shoulder label specifies a vehicle standing on the shoulder lane of the highway.
- **Traffic Jam:** With a traffic jam present, the chance of finding an accident is pretty high.
- **Breakdown:** Breakdowns on the shoulder lane and especially on a driving lane, can foreshadow accidents on the highway.
- **Accident:** The accident label shows all accidents, which were found in the recordings from the A9 test stretch.

# Nr.	Aa Date	☰ Event	☰ Description
1	📄 13.10.2020	Traffic Jam	Traffic jam on the highway exit
2	📄 24.01.2021	Emergency Vehicle Vehicle transport	2 emergency vehicles, 10 vehicle transports and the accident of event 4
3	📄 11.02.2021	Vehicle transport Emergency Vehicle	Emergency vehicle (not in operation) and 2 vehicle transporters
4	📄 08.04.2021	Accident	Yellow car loses control and crashes first into crash barrier and then into standing white van
5	📄 15.05.2021	Shoulder Vehicle transport Emergency Vehicle	6 times standing vehicle on hard shoulder, vehicle transporter and emergency vehicle (not in operation)
6	📄 29.07.2021	Shoulder	Standing vehicle on hard shoulder
7	📄 30.07.2021	Shoulder	Standing vehicle on hard shoulder
8	📄 30.07.2021	Traffic Jam	Traffic jam in south direction
9	📄 21.10.2021	Breakdown	Blue VW bus tips over because it was too windy
10	📄 07.03.2022	Emergency Vehicle Tow truck	Emergency vehicle (not in operation) & tow truck
11	📄 28.03.2022	Accident	Vehicle runs into the end of a traffic jam and collides with 2 other vehicles (360 degree spin)
12	📄 28.03.2022	Emergency Vehicle	Emergency vehicle after the accident
13	📄 11.05.2022	Vehicle transport Tow truck Emergency Vehicle	Lots of vehicle transporters, a few emergency vehicles as well as a tow truck
14	📄 22.05.2022	Shoulder Breakdown	Vehicle burns on hard shoulder

**Figure 4.1:** Overview of the Event Log.

From the Event Log the following four traffic situations were used to construct the rules in Subsection 4.2:

(a) Accident event on the 08.04.2021:

Figure 4.2 illustrates the time series of the accident on the 08.04.2021, which was used to fine-tune the accident detection in Section 4.2.2.4. The cause of the accident was the



(a) Illustration of the traffic situation before the accident. (b) Illustration of the traffic situation where the yellow vehicle drove into the guard rail.



(c) Illustration of the traffic situation where the yellow vehicle drove into the white van. (d) Illustration of the traffic situation after the accident.

**Figure 4.2:** Time series for the accident event on the 08.04.2021, where a yellow vehicle crashed into a standing white van on the highway.

white van on the rightmost lane of the left side of the highway, which had a breakdown and therefore had to stop on a driving lane. The yellow vehicle was driving 180 km/h and lost control of the vehicle whilst trying to evade crashing into the white van, which is visible in Figure 4.2(a). In Figure 4.2(b), the yellow vehicle crashed into the guard rail and still had enough momentum to crash into the white van in Figure 4.2(c). Figure 4.2(d) then shows the end of the accident, where both vehicles are standing still.

(b) Accident event on the 21.10.2021:

Figure 4.3 depicts the time series of a breakdown event on the 21.10.2021 on the left side of the highway. Whilst Figure 4.3(a) illustrates the traffic situation right before the breakdown, Figure 4.3(b) shows the moment in which the trailer of the blue van is about to be knocked over by the wind. In Figure 4.3(c) the trailer fell over completely, leading to the blue van being knocked over as well as shown in Figure 4.3(d).

(c) Accident event on the 28.03.2022:

Figure 4.4 illustrates the second recorded crash between multiple vehicles. Figure 4.4(a) depicts the traffic situation right before the accident occurred. Due to a traffic jam, the taxi driver is slowing down on the farthest left lane of the right highway side. The black vehicle behind the taxi realizes this too late, evades to the right lane and crashes into the blue vehicle in Figure 4.4(b). Afterward, the black vehicle also crashes into the taxi



(a) Illustration of the traffic situation before the breakdown.



(b) Illustration of the traffic situation where the trailer of the blue van is about to fall over.



(c) Illustration of the traffic situation after the trailer of the blue van fell over.



(d) Illustration of the traffic situation after the trailer and the blue van fell over.

**Figure 4.3:** Time series for the breakdown on 21.10.2021, where a blue van was knocked over by the wind.



(a) Illustration of the traffic situation before the accident.



(b) Illustration of the traffic situation where the black vehicle drove into the side of the blue vehicle.



(c) Illustration of the traffic situation where a black vehicle drove into the taxi.



(d) Illustration of the traffic situation after the accident.

**Figure 4.4:** Time series for the accident event on the 28.03.2022, where a black vehicle is driving into the side of a blue vehicle and spinning it 360 degrees.

as shown in Figure 4.4(c). While the black vehicle and the taxi are coming to a standstill after the crash, the blue vehicle does a 360-degree spin before it stops moving in Figure 4.4(d).

(d) Breakdown event on the 22.05.2022:

Figure 4.5 shows a breakdown event on the 22.05.2022, where a vehicle on the left shoulder lane burned down and afterward got transported away by a truck. While the vehicle looks completely fine in Figure 4.5(a), a few minutes later in figure 4.5(b) the vehicle is smoking. Figure 4.5(c) then shows the vehicle burning down and omitting even more smoke. Lastly in Figure 4.5(d), the vehicle burnt down completely and a vehicle transporter arrived to transport the burning vehicle away safely.



(a) Illustration of the traffic situation before the vehicle burns.



(b) Illustration of the traffic situation where the vehicle starts to smoke.



(c) Illustration of the traffic situation where the vehicle is burning.



(d) Illustration of the traffic situation after the vehicle burned down.

**Figure 4.5:** Time series for the breakdown event on the 22.05.2022, where a burning vehicle is standing on the shoulder lane.

## 4.2 Scenario Mining

The aim of the scenario mining is to create an automated approach, which labels the input scenarios into different types. The input scenarios in this case are the recorded rosbags from the A9 test stretch. Firstly, the raw recordings from the rosbags have to be extracted into data frames to then apply the feature extraction and afterward the maneuver detection. The extraction of the raw recordings, the lane ID as well as the distance to leading and following

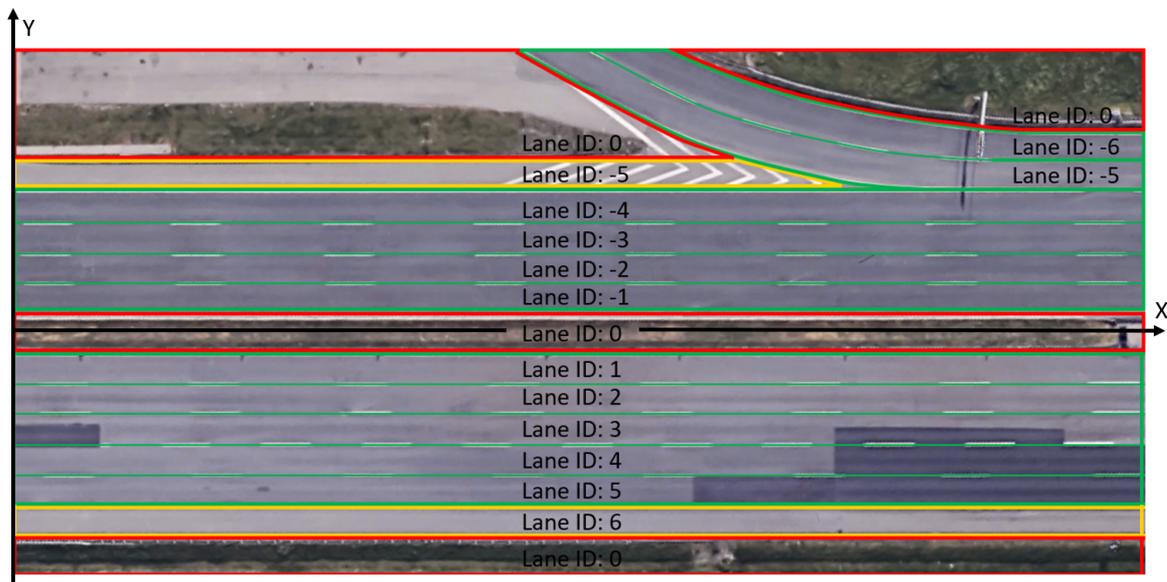
vehicle feature extraction and the cut in/out, speeding, standing, and tailgate maneuver detection has already been implemented by Aaron Kaefer [1]. Therefore, this section follows the previous work of Aaron Kaefer closely and improves as well as extends his work of numerous features and maneuvers. However, this approach strongly depends on accurate input data such as the continuous detection of vehicles, their 3D position estimation as well as the speed, which comes from a perception pipeline using YOLOv4 for object detection and object tracking.

#### 4.2.1 Feature Extraction

The feature extraction is responsible for extracting all features, which are necessary for the maneuver detection, from the extracted raw recordings of the rosbags. Necessary features for accident detection are the lane ID in which the vehicles are driving, the corresponding distance to the leading and following vehicle in the respective lane as well as the average velocity on both sides of the highway.

##### 4.2.1.1 Feature: Lane ID

To detect maneuvers such as standing vehicles on the shoulder lane and rear-end collisions, the lane ID of each vehicle in every frame has to be extracted from the raw recordings. Figure 4.6 shows a section of the A9 test stretch from the bird's eye view. The values for the different lane IDs are distributed according to the OpenDRIVE format, meaning the lane between the highway sides has lane ID 0. The lanes towards the bottom range from 1 to 6 and the lanes towards the top range from -1 to -6.



**Figure 4.6:** Illustration of the different lane IDs on the highway.

The lanes are split up into 3 different kinds:

- (a) The driving lanes are visualized in green.
- (b) The shoulder lanes are visualized in yellow.

(c) The boulevards are visualized in red.

Instead of Kaefer's approach [1], where one polygon was used to model one lane and afterward it was checked if the center of the vehicles' bounding box was inside one of the polygons, a faster approach was needed to process as many rosbags in a short period of time as possible. Therefore, the proposed solution only checks the y position of the vehicle's center of the bounding box to label the lane ID, since this avoids computational heavy checks e.g. if a polygon contains a specific point. Taking the center of the bounding box of each vehicle is crucial to circumvent any ambiguous lane ID labels in cases where the vehicle is driving in multiple lanes simultaneously, which would lead to unwanted behavior.

Figure 4.7 visualizes the same section of the A9 test stretch with additional vehicles. In this case, the vehicles with ID 1, 2 and 3 would get the label 4 for lane ID and vehicles 4, 5 and 6 the label -1 for lane ID. Since the center of the bounding box is used for labeling, vehicle 2 only gets label 4 instead of having label 4 and label 5, even though the vehicle is driving in both lanes.

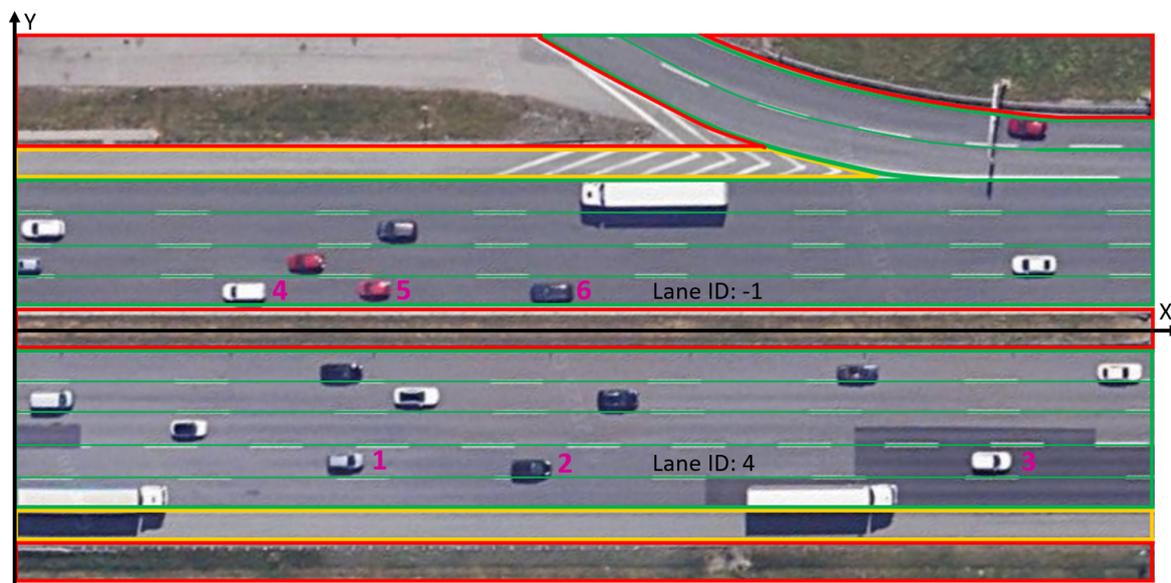


Figure 4.7: Example of the lane ID labeling.

#### 4.2.1.2 Feature: Distance to Leading and Following Vehicle

The distance to the leading and following vehicle is crucial to detect multiple maneuvers such as cut-ins and cut-outs and especially accidents. While a possible solution was already proposed by Kaefer [1], the computation time was not scalable for bigger projects. Therefore, this section focuses on a new approach.

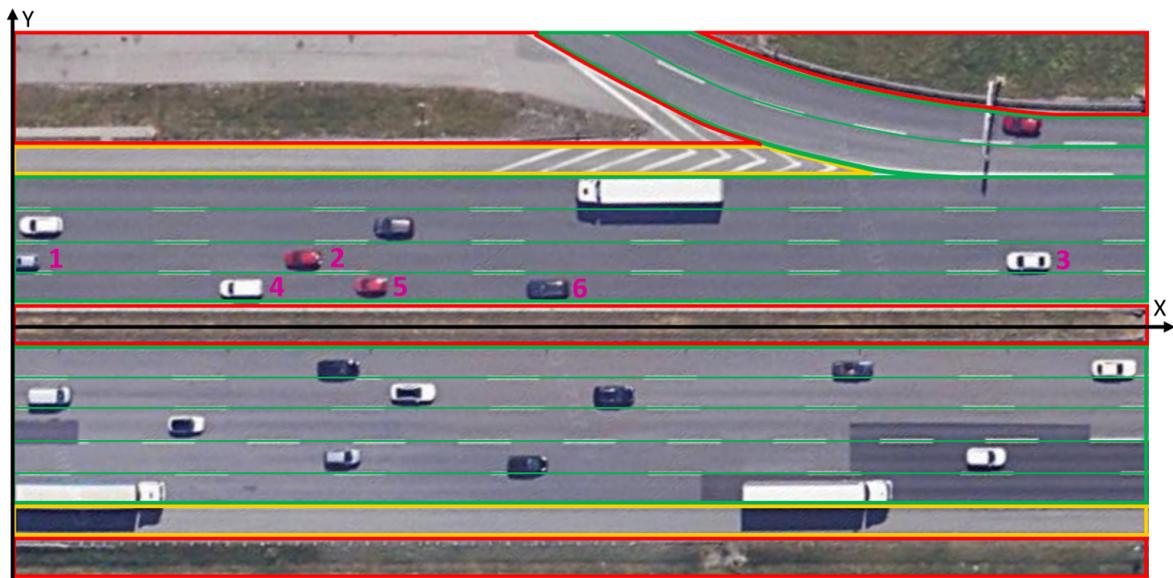
1. Firstly, all traffic participants are being split up according to their lane ID. Whilst this does limit the use case, it speeds up the computation time drastically.
2. Secondly, the traffic participants in each lane are sorted by their x value of the bounding box center.
3. Then the Euclidean Distance Matrix (EDM) between neighboring traffic participants is calculated and stored as distance to the leading and following vehicle respectively.

Neighboring traffic participants are the vehicles at index 0 and index 1, index 1 and index 2 all the way to index N-1 and index N.

This approach achieves two significant speed-ups compared to Kaefer's approach:

1. Instead of computing the EDM between every traffic participant in a lane, it is now only being computed for the necessary neighboring vehicles.
2. Instead of interpolating the driven trajectory of two vehicles and calculating the corresponding Euclidean distance, the new approach takes the location from the raw recordings and only calculates the EDM, which does not take the square root in the end, unlike the euclidean distance.

Figure 4.8 visualizes an example of the distance to leading and following vehicle calculation. There are three traffic participants currently driving in lane ID -2. Instead of calculating the EDM for every possible traffic participant permutation, which would be (1,2), (2,3), and (1,3), it is computationally faster to sort their x value and only calculate the EDM for (1,2) and (2,3). While this does not make a noticeable difference in this case, the computational cost grows exponentially with the amount of traffic participants in the corresponding lane. A disadvantage to this approach is the fact that vehicle 2 and vehicle 5 are driving relatively close to one another and since they do not share a lane, their respective distance is not being calculated. However, in this solution, the computational overhead for calculating the EDM in such cases would exceed the scalability.



**Figure 4.8:** Example of the distance to leading and following vehicle calculation.

#### 4.2.1.3 Feature: Average Velocity / Average Velocity per Frame

Additionally, to the lane ID and distance to leading and following vehicle, the average velocity as well as the average velocity per frame is necessary to detect maneuvers such as traffic jams, slow-moving traffic, and breakdowns on a driving lane:

- Average Velocity:

The average velocity on the highway is split up into two different categories:

- `average_velocity_north`: average velocity on the side driving towards the northern direction
- `average_velocity_south`: average velocity on the side driving towards the southern direction

The calculation is then done as follows:

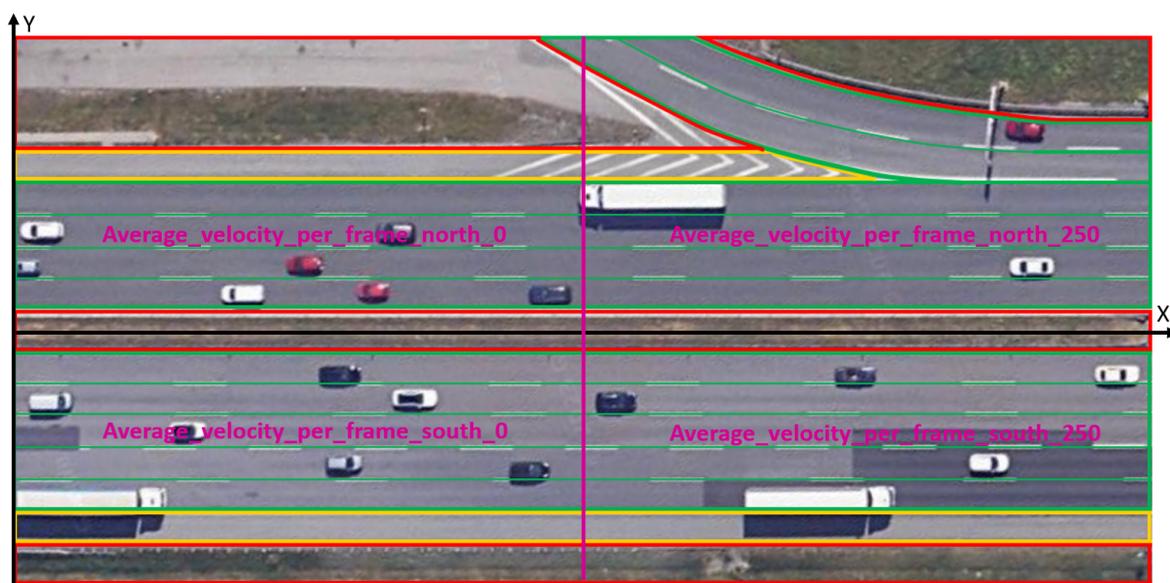
1. Fetch all velocities of each frame and store them corresponding to which side the traffic participant is driving on
2. Calculate the mean of the velocity for both highway sides

Both values are important for the scenario statistics, which is going to be covered in Section 4.2.3, since the average velocity on a side of the highway is directly correlated to traffic jams, slow-moving traffic, and potential accidents.

- Average Velocity per Frame:

Figure 4.9 illustrates the split of the average velocity per frame on the highway into four different categories:

- `average_velocity_per_frame_north_0`: average velocity per frame on the side driving towards the northern direction and in the x value interval [0, 250)
- `average_velocity_per_frame_north_250`: average velocity per frame on the side driving towards the northern direction and in the x value interval [250, 500]
- `average_velocity_per_frame_south_0`: average velocity per frame on the side driving towards the southern direction and in the x value interval [0, 250)
- `average_velocity_per_frame_south_250`: average velocity per frame on the side driving towards the southern direction and in the x value interval [250, 500]



**Figure 4.9:** Illustration of the split of the average velocity per frame into four different categories.

The calculation is then done as follows:

1. Iterate over each frame and fetch for each frame all velocities of the single frame. Store them corresponding to which of the four categories the traffic participant is currently driving on
2. Calculate the mean of the velocity for the specific frame for all four categories

The average velocity per frame values are needed to calculate the traffic jams, slow-moving traffic as well as breakdowns on driving lanes in Section 4.2.2.

## 4.2.2 Maneuver Detection

The goal of the maneuver detection is to detect maneuvers on the highway with the help of the extracted features from Section 4.2.1. Possible maneuvers are cut-ins, cut-outs, standing events, breakdowns, traffic jams, and most importantly accidents.

### 4.2.2.1 Label: Standing Shoulder

Since in Figure 4.6 the differences between driving lanes, shoulder lanes, and boulevards were already illustrated in detail and Kaefer [1] implemented a label for standing traffic participants, the approach for the standing shoulder label was straight forward:

1. Check if any traffic participant has either lane ID 6 or lane ID -5, but is not on the highway exit, which can be done by checking the x value of the traffic participant.
2. Check if the traffic participant from above has a velocity below 0.04, which was used as a threshold by Kaefer [1] to determine standing vehicles.
3. If both conditions above are met, classify the traffic participant as a standing shoulder vehicle.

### 4.2.2.2 Label: Traffic Jam / Slow-moving traffic

While there are many different definitions of traffic jams and slow-moving traffic on the highway, the label for traffic jams as well as for slow-moving traffic takes this definition<sup>1</sup> as the baseline:

- Traffic jam: Throughout 1 km, multiple traffic participants have to drive below 20 km/h for at least 5 minutes.
- Slow-moving traffic: Throughout 1 km, multiple traffic participants have to drive between 20 km/h and 40 km/h for at least 5 minutes.

Due to the characteristics of the A9 test stretch, the following changes have to be made:

- Since the A9 test stretch is only 500 m long, the stretch for traffic jams and slow-moving traffic is being reduced to 500 m.
- Since multiple recorded rosbags are only 1 minute long, the time constraint is being lowered from 5 minutes to 30 seconds.

<sup>1</sup><https://www.leasingmarkt.de/magazin/reisen/stau-tipps#:~:text=Um%20einen%20Stau%20handelt%20es,man%20hingegen%20von%20stockendem%20Verkehr.>

- Since the average velocity per frame was already calculated in Section 4.2.1.3, instead of checking if multiple traffic participants drive below 20 km/h or 40 km/h the average velocity per frame has to be below the mentioned threshold.

The implementation for traffic jams and slow-moving traffic is done as follows:

1. Execute the traffic jam and slow-moving traffic calculation for both the northern and southern sides of the highway.
2. Since the definition of a traffic jam and slow-moving traffic specifies that the average velocity per frame throughout 500 m has to be analyzed, it is necessary to split the stretch into multiple parts, which are being examined separately. Therefore, fetch the average velocity per frame for the corresponding highway side in the interval [0, 250] and in the interval [250, 500].
3. Check if both average velocities per frame are either below 20 km/h (traffic jam) or between 20 km/h and 40 km/h (slow-moving traffic).
4. Check if the average velocity per frame fulfills the same case for at least 30 seconds.
5. Classify if a traffic jam or slow-moving traffic occurred.

#### 4.2.2.3 Label: Breakdown

Additionally, to traffic jams and slow-moving traffic, breakdowns can be closely correlated to accidents as already shown in Figure 4.2, where the white van broke down on a driving lane causing an accident.

- Breakdown shoulder:

Most commonly, the traffic participant having the breakdown changes lanes towards the shoulder lane and stops there. Therefore, all traffic participants standing for at least 30 s on a shoulder lane get the breakdown shoulder label. The implementation is done as follows:

1. Iterate over each actor and check the standing shoulder label in each frame.
2. If the vehicle stands for at least 30s on the shoulder lane, it is being labeled as a breakdown shoulder event

- Breakdown driving lane:

In some rare cases, the breakdown vehicle can not change lanes towards the shoulder lane and stops on a driving lane, causing the accident probability to increase drastically. Therefore, all traffic participants standing for at least 30s on a driving lane, while there is no traffic jam, are being labeled as a breakdown driving lane.

1. Iterate over each actor and check if the traffic participant is standing, but not on a shoulder lane.
2. Check if the average velocity per frame on the corresponding side of the highway and in the interval of the traffic participant itself is above 20 km/h to avoid FP detections during traffic jams.
3. If the condition above is met for at least 30 s, it is being labeled as a breakdown driving lane event.

#### 4.2.2.4 Label: Accident

The main contribution of this paper is a rule-based approach for accident detection on the recorded data from the A9 test stretch in order to create an anomaly traffic dataset, which can be used with a more sophisticated model in the future, which achieves a higher precision and recall. As of now the rule-based accident detection is limited to finding crashes between two vehicles, which are driving in the same lane. After careful consideration and analysis of the recorded accident on the 08.04.2021 shown in Figure 4.2, the following rules were established:

$$velocity_i \geq \frac{15 \text{ km/h}}{3.6} \quad (4.1)$$

$$velocity_i > velocity\_lead_i \quad (4.2)$$

$$velocity_i \geq velocity_j \quad \forall \quad i < j \leq N \quad (4.3)$$

$$distance\_lead_i \geq distance\_threshold \quad (4.4)$$

$$distance\_lead_i < \left( \frac{velocity_i - velocity\_lead_i}{30} \right)^2 \quad (4.5)$$

$$ttc\_leading_i \leq ttc\_threshold \quad (4.6)$$

The rules are responsible for calculating if a specific traffic participant was involved in an accident. It is done by iterating through each frame the actor was recorded in and checking the respective values for that frame. Let  $i$  be the frame, which is currently being checked for accidents, and  $N$  the number of frames the specific traffic participant was in the range of the sensors.

1. Firstly, in Equation 4.1 the velocity of the traffic participant has to be at least 15 km/h. This is done to filter out potential FP in traffic jams or slow-moving traffic scenarios since the object detection used in the perception pipeline is not capable of estimating the 3D position of the traffic participants accurately.
2. Secondly, in Equation 4.2 the velocity of the traffic participant has to be greater than the velocity of the vehicle in front, because otherwise the traffic participant in the back is not going to catch up with the one in front and is therefore not going to cause an accident.
3. Equation 4.3 adds the constraint that the velocity of the traffic participant can not increase after the accident, since it is unusual to continue driving after a crash.
4. Equation 4.4 ensures that the distance to the leading vehicle is higher than a certain threshold. This is necessary, because the digital twin was created using YOLOv4. Therefore, the object detection from YOLO detects multiple FP traffic participants. The bounding boxes of these FP and the actual traffic participants often overlap or are relatively close to each other. After fine-tuning the `distance_threshold`, the most promising results on the rosbags covered in the event log were achieved with 0.1.
5. Equation 4.5 defines the constraint for the distance between the two accident traffic participants. Since the distance, which should be kept between vehicles on the highway, is correlated to both the own velocity and the velocity of the leading traffic participant, the  $\Delta velocity$  is used to check the distance. Additionally, the  $\Delta velocity$  is divided by 30, since this yielded the best result on the rosbags of the event log. Last but not least, the term on the right side needs to be squared, because the distance to the leading vehicle was calculated using EDM as covered in Section 4.2.1.2, which means the square

root was neglected while calculating the euclidean distance.

For a better understanding of the distance constraint, Equation 4.7 is being transformed into Equation 4.8 by squaring both sides. As already covered above, taking the square root of the distance to the leading vehicle is the same as calculating the Euclidean distance between the (x,y) points of the two traffic participants. Since the value of the Euclidean distance is always positive as well as the right side of Equation 4.7, which is both due to the square and Equation 4.2, the transformation is mathematically correct.

$$distance\_lead_i < \left( \frac{velocity_i - velocity\_lead_i}{30} \right)^2 \quad (4.7)$$

$$\|point - point\_lead\|_2 < \left( \frac{velocity_i - velocity\_lead_i}{30} \right) \quad (4.8)$$

#### Example of the accident on 08.04.2021:

If actor1 drives 33 m/s and actor2 drives 0 m/s, then as shown in Equation 4.9 once the distance between the two traffic participants is below 1.1 an accident is classified according to the rule. In the real accident, the distance between the two vehicles was 0.75 m.

$$threshold = \frac{33 - 0}{30} = 1.1m \quad (4.9)$$

6. Additionally, Equation 4.6 checks whether the Time-to-collision (TTC), which was already implemented by Kaefer [1] is below the `ttc_threshold` being set to 0.1.

If and only if all six rules apply simultaneously, the corresponding traffic participant is classified as an accident event.

#### 4.2.3 Scenario Statistics

After the feature extraction and the maneuver detection are finished, a statistical summary is created and stored in a JavaScript Object Notation (JSON) file for every executed rosbag. The statistics contain the information of all maneuvers, which occurred in the rosbag such as standing vehicles on the shoulder lane, traffic jams, slow-moving traffic, and accidents as well as the statistics, which were already implemented by Aaron Kaefer [1]. Table 4.1 shows all statistics, which were added in this paper.

### 4.3 Automated Accident Detection

With the working rule-based approach for accident detection, an automated approach for the accident detection on the rosbags of multiple years, which were recorded on the A9 test stretch and are stored online in a cloud storage, can be developed. The automated accident detection consists of the following steps:

1. Firstly, list all directory contents from the remote cloud storage and store them in a list.

Statistics	Description
Total Vehicles	Sum of all detected vehicles
Total Vehicle Classes	Amount of different detected vehicle classes, e.g. such as car, bus, truck, motorcycle etc.
Total Standing Vehicles Shoulder	Sum of all detected standing shoulder vehicles
Average Velocity North	Average Velocity of all detected vehicles on the northern highway side
Average Velocity South	Average Velocity of all detected vehicles on the southern highway side
Traffic Jam North	Flag specifying if a traffic jam occurred on the northern highway side
Traffic Jam South	Flag specifying if a traffic jam occurred on the southern highway side
Slow-moving Traffic North	Flag specifying if there is slow-moving traffic on the northern highway side
Slow-moving Traffic South	Flag specifying if there is slow-moving traffic on the southern highway side
Total Breakdowns Shoulder	Sum of all detected breakdowns on a shoulder lane
Total Breakdowns Driving Lane	Sum of all detected breakdowns on a driving lane
Total Breakdowns	Sum of all detected breakdowns
Total Accidents	Sum of all detected accidents

**Table 4.1:** Overview of the statistics added to Kaefers [1] scenario statistics.

2. Iterate over the contents and check the following conditions for every content:
  - (a) Check if the content is a rosbag file.
  - (b) Check if the rosbag file takes place after the `download_start` value, which specifies a timestamp for which the download of the rosbag should start. For example if `download_start="/year_2021/month_12/day_20"`, then all rosbags taking place before the 20.12.2021 are not being downloaded and therefore the accident detection is not being executed for them. This helps to skip rosbags, which were already executed in earlier stages.
  - (c) Check if the rosbag takes place during a specific time interval. In this paper due to limited time, the time interval is chosen according to the sunrise and sunset in each month, because the object detection is not working as accurately in rosbags, which take place during the night.
3. If all three conditions above are being met, the rosbag file is then downloaded from the cloud storage.
4. Since the execution of the scenario mining for the recorded data on an entire year is not scalable, a separate script is being executed on the downloaded rosbag, which checks if a standing vehicle is detected. This will filter all anomalous rosbags, since in all important cases such as traffic jams, breakdowns, and accidents, at least one standing vehicle has to be present.
5. Next, the scenario mining is being executed on all rosbags, where a standing vehicle was detected in the step before. The scenario statistics are being calculated for every rosbag

in this step to analyze all standing shoulder vehicle events, traffic jams, slow-moving traffic, breakdowns, and accidents.

6. In the end, the rosbag is being deleted, since the used Solid State Drive (SSD) is not big enough to store all 200,000 recorded rosbags from the A9 test stretch, which takes up a total of 100 TB.



# Chapter 5

## Evaluation

This work proposes a simple rule-based approach for accident detection on the A9 test stretch on the highway. This approach is included in an automated accident detection script to find accidents in large datasets. With the help of the automated accident detection and other labels such as standing shoulder vehicles, traffic jams, slow-moving traffic and breakdowns an anomaly traffic dataset can be created to train a more sophisticated deep learning model for accident detection, which achieves a higher precision and recall rate. In this chapter, the main contributions, which consist of the scenario mining and afterward the automated accident detection, are being evaluated based on the two research questions if it is possible to reliably detect accidents on the A9 test stretch using roadside sensors and if this procedure can be executed in real-time.

### 5.1 Metrics

To evaluate both the scenario mining approach with all its features and labels as well as the automated accident detection, the following metrics are being used:

- **Scenario Mining**

Firstly, the scenario mining approach is being evaluated based on the achieved speedup and its functional correctness.

- **Speedup:** With the automated accident detection being used to analyze big datasets, keeping the computational performance of the scenario mining as low as possible is crucial for real-world applicability. The speedup is defined as follows, where the execution time of Kaefer's approach  $t_K$  is divided by the execution time of the proposed approach  $t_P$ :

$$speedup = \frac{t_K}{t_P} \quad (5.1)$$

Additionally, the speedup can be split up into the speedup of the lane ID extraction shown in Equation 5.2 and the speedup of the distance extraction, shown in Equation 5.3, for a more detailed analysis:

$$speedup\_lane\_ID = \frac{t_{K\_laneID}}{t_{P\_laneID}} \quad (5.2)$$

$$speedup\_distance = \frac{t_{K\_distance}}{t_{P\_distance}} \quad (5.3)$$

- **Functional Correctness:** Before evaluating the automated accident detection, the functional correctness of the labels in the maneuver detection has to be ensured. Therefore, the scenario mining is being evaluated on the four rosbags, which were already explained in Section 4.1.

- **Automated Accident Detection**

After evaluating the real-world applicability of the scenario mining and its functional correctness, the whole automated accident detection procedure can be evaluated on unseen data from the A9 test stretch. The automated accident detection is evaluated based on precision and a false negative analysis.

- **Precision:** The precision is widely used to evaluate how often a machine learning model correctly predicts the positive class [9] in correlation to all positive class predictions. Equation 5.4 shows the definition for the precision.

$$precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \quad (5.4)$$

- **False Negative Analysis:** Since the scenario mining stores the scenario statistics for every executed rosbag, the occurred breakdowns can be analyzed manually to detect potential false negatives in the accident detection.

- **Comparison with Deep Learning-based Accident Detection**

After evaluating the automated accident detection using the rule-based approach, it can be compared to a deep learning-based accident detection, which was implemented by Daniel Lehmborg in parallel to this work.

- **Precision:** After evaluating the precision on the recorded data from the A9 test stretch, the precision of both approaches is being compared on a manually created test set.
- **Recall:** Additionally to the precision, the recall of both approaches is being compared. The recall defines how often the model correctly classifies positive instances from all positive instances. Equation 5.5 shows the definition for the recall.

$$recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \quad (5.5)$$

- **Runtime:** Last but not least, the runtime for both approaches is being analyzed to make a meaningful statement about the scalability of both approaches.

## 5.2 Structure

To evaluate the contribution with the metrics mentioned in Section 5.1, Lehmborg implemented an image extraction in the scenario mining step. Additionally, a deep learning-based accident detection was written for the comparison with the rule-based accident detection, which was trained on frames of the available accidents as well as online data<sup>1</sup>.

### 5.2.1 Image Extraction

To evaluate the labels in the scenario mining, it was extended of an image extraction, which is responsible for extracting the images for specific labels. The following labels can be passed as command line arguments:

- **standing\_shoulder:** Extract the images for each vehicle, which stands on the shoulder lane.
- **breakdown:** Extract the images for each vehicle, which has a breakdown.
- **accident:** Extract the images for each detected accident.
- **accident\_breakdown:** Extract the images for each vehicle, which has a breakdown as well as for each detected accident.

To speed up the process of the image extraction only three images are being extracted for a single event. The image 4.8 seconds before the event, the image of the frame, where the event occurred, and the image 4.8 seconds after the event, since this is enough to evaluate the event correctly.

Figure 5.1 depicts the output folder structure if four rosbags are being executed and the accident command line argument is set. If a rosbag contains an event, for which the image extraction was enabled, a folder is being created for this rosbag. In Figure 5.1 three different accidents were detected, two in rosbag\_0 and one in rosbag\_3. Therefore, a folder was created for both rosbags. Each detected event is being stored in a separate folder to differentiate between multiple events in a single rosbag. As a consequence, rosbag\_0 has two folders for the two detected accidents, and rosbag\_3 has only one folder. Each event folder then consists of four folders, which represent the four cameras and contain the three extracted images of the corresponding camera each. Additionally, the calculated scenario statistics are stored as a JSON file in the folder of the corresponding rosbag. Since the statistics for the rosbags, which do not have the specified event such as an accident, are still important to analyze the data, they are stored in a separate statistics folder. Therefore, in Figure 5.1 the statistics for rosbag\_1 and rosbag\_2 are being stored in the statistics folder.

In the evaluation, the scenario mining is executed with the `accident_breakdown` command line argument. The `accident` value is necessary to evaluate the accidents and the `breakdown` value helps to find potential false negatives in the accident detection since accidents have a high occurrence probability in scenarios with breakdowns.

---

<sup>1</sup><https://universe.roboflow.com/accident-detection-model/accident-detection-model/dataset/2>

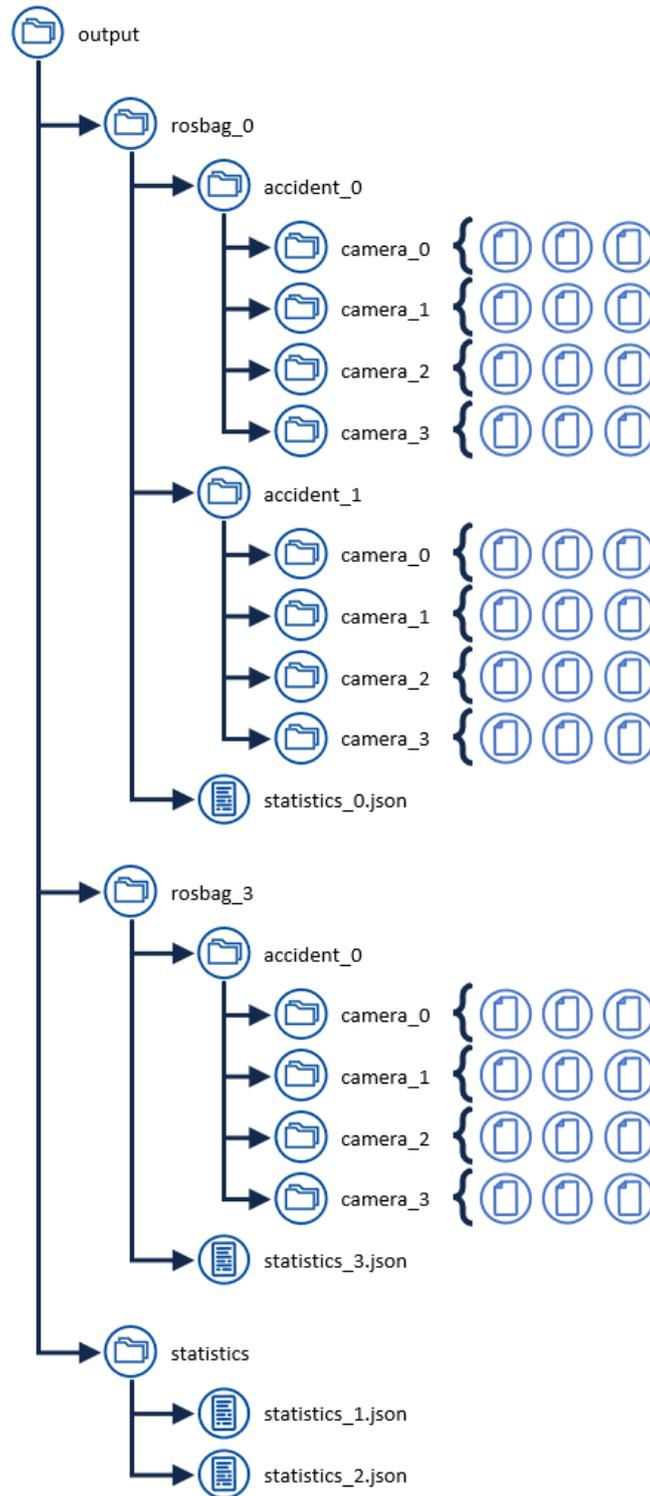


Figure 5.1: Illustration of the folder structure of the image extraction.

### 5.2.2 Deep Learning-based Accident Detection

Since state-of-the-art accident detections mostly use deep learning-based models, Daniel Lehmborg implemented such an approach for the project, which uses images as input to detect

accidents. Firstly, the YOLOv8x<sup>2</sup> model was used as the baseline model, which was pre-trained on Common Objects in Context (COCO). COCO is a large-scale dataset for the use case of object detection, segmentation, and captioning from Microsoft [19]. To detect accidents, the pre-trained YOLOv8x model was then fine-tuned on a dataset consisting of manually labeled accident images from the recorded data and accident images, which were already used for the training process of other deep learning-based accident detection models<sup>3</sup>.

## 5.3 Results

This section illustrates and analyzes the measured results for the mentioned metrics from Section 5.1. The results were measured on a system using Ubuntu 20.04.4 Long Term Support (LTS) with an Intel Core i9-9900KF Central Processing Unit (CPU), 3.6 GHz and 8 cores, 32 GB Random Access Memory (RAM) and a NVIDIA Corporation TU104 [GeForce RTX 2080 SUPER] graphics card.

### 5.3.1 Scenario Mining

#### 5.3.1.1 Speedup

To analyze as many rosbags as possible in a short time period, improving the computational performance of the scenario mining is crucial for the real-world applicability.

Table 5.1 illustrates the average runtime of the scenario mining for both Kaefer's approach [1] and the approach proposed in this paper. The scenario mining was executed 20 times for the 15-minute rosbag on the 22.05.2022 and the runtime values were averaged.

Since the scenario extraction was not changed in this paper, the average runtime is approximately the same for Kaefer's approach and the proposed approach with a runtime difference of 0.6 s. However, the lane ID extraction was reduced from 937.49 s from Kaefer's approach [1] to only 2.22 s in the proposed approach from Section 4.2.1.1. This leads to a lane ID extraction speedup of 422.29 as shown in Equation 5.6:

$$speedup_{lane\_ID} = \frac{937.49 \text{ s}}{2.22 \text{ s}} = 422.29 \quad (5.6)$$

In addition to the speedup of the lane ID extraction, the distance to leading and following vehicle extraction has also been improved significantly. While Kaefer's approach [1] needs 3371.73 s to extract the distance values, the proposed approach in Section 4.2.1.2 only takes 178.02 s leading to a speedup of the distance to leading and following vehicle extraction of 18.94 as shown in Equation 5.7:

$$speedup_{distance} = \frac{3371.73 \text{ s}}{178.02 \text{ s}} = 18.94 \quad (5.7)$$

Due to additional features such as the average velocity extraction and labels such as traffic jams, slow-moving traffic, breakdowns, and accidents, the runtime of the average velocity

<sup>2</sup><https://docs.ultralytics.com/de/tasks/detect/>

<sup>3</sup><https://universe.roboflow.com/accident-detection-model/accident-detection-model/dataset/2>

extraction, maneuver detection, accident detection, and the statistics creation proposed in this paper is higher than Kaefers [1] runtime. All in all, the scenario mining approach for a 15-minute rosbag takes now instead of 4403.27 s only 234.25 s even with the additional features and labels. Therefore, an overall speedup of 18.8 was achieved with this paper as shown in Equation 5.8:

$$speedup = \frac{4403.27 \text{ s}}{234.25 \text{ s}} = 18.80 \quad (5.8)$$

	Kaefers approach	Proposed approach
Scenario Extraction	46.25s	46.85s
Lane ID Extraction	937.49s	2.22s
Distance to Leading / Following Vehicle Extraction	3371.73s	178.02s
Average Velocity Extraction	not implemented	4.38s
Maneuver Detection	0.59s	0.92s
Accident Detection	not implemented	0.15s
Statistics Creation	0.40s	0.53s
Overall Runtime	4403.27s	234.25s

**Table 5.1:** Average performance comparison of Kaefers approach with the approach proposed in this paper using the 15-minute rosbag on the 22.05.2022.

### 5.3.1.2 Functional Correctness

For the automated accident detection to work properly, the added labels such as standing shoulder vehicles, breakdowns, slow-moving traffic, traffic jams as well as accidents have to be functionally correct. Therefore, the functional correctness is being analyzed on the four rosbags from section 4.1:

- Accident event on the 08.04.2021:

Listing 5.1 shows the JSON containing the scenario statistic of the minute right before the accident on the 08.04.2021, which was already illustrated in Figure 4.2. In the minute before the accident, there was one standing vehicle in a driving lane, which was the white van standing on the farthest right lane of the left highway side. Additionally, this vehicle was also classified as a breakdown vehicle on a driving lane, which ensures the functional correctness of the breakdown label.

```
{
  "total_vehicles": 245,
  "total_vehicle_classes": 3
  "total_lane_changes_left": 17,
  "total_lane_changes_right": 75,
  "total_lane_changes": 92,
  "max_lane_changes": 3,
  "total_cut_ins_left": 0,
  "total_cut_ins_right": 1,
  "total_cut_ins": 1,
  "total_cut_outs_left": 1,
  "total_cut_outs_right": 1,
```

```

"total_cut_outs": 2,
"total_tail_gates_1": 37,
"total_tail_gates_2": 10,
"total_tail_gates_3": 5,
"total_speeding_vehicles": 32,
"total_standing_vehicles": 1,
"total_standing_vehicles_shoulder": 0,
"top_speed": 49.76776123046875,
"average_velocity_north": 16.591909942654368,
"average_velocity_south": 30.73570799615174,
"traffic_jam_north": 0,
"traffic_jam_south": 0,
"slow_moving_traffic_north": 0,
"slow_moving_traffic_south": 0,
"total_trajectories": 245,
"total_breakdowns_shoulder": 0,
"total_breakdowns_driving_lane": 1,
"total_breakdowns": 1,
"total_accidents": 0
}

```

**Listing 5.1:** Scenario statistics of the minute before the accident rosbag on the 08.04.2021

Listing 5.2 illustrates the JSON file for the accident on the 08.04.2021, which occurred in second 13 of 60. While the average velocity on the southern side is almost 29 m/s, the average velocity on the northern side is only 12 m/s. However, the average velocity was at around 17 m/s a minute before hinting at the possibility of a potential accident. Since the rosbag is only 1 minute long, there was not enough time to form a traffic jam or slow-moving traffic on the whole 500 m of the test stretch. However, the average velocity on the northern side is already fairly close to slow-moving traffic. Additionally, 43 vehicles were standing out of all 301 total detected vehicles in the rosbag, which is also an indication of an accident. Moreover, no breakdowns were detected since the crash already occurred in second 13 and the object detection from YOLOv4 was not able to detect the two accident vehicles continuously after the crash. In addition, the accident was detected making the rule-based approach applicable to find new accidents in the recorded rosbags.

```

{
  "total_vehicles": 301,
  "total_vehicle_classes": 3,
  "total_lane_changes_left": 8,
  "total_lane_changes_right": 50,
  "total_lane_changes": 58,
  "max_lane_changes": 3,
  "total_cut_ins_left": 0,
  "total_cut_ins_right": 0,
  "total_cut_ins": 0,
  "total_cut_outs_left": 0,
  "total_cut_outs_right": 0,
  "total_cut_outs": 0,
  "total_tail_gates_1": 30,
  "total_tail_gates_2": 8,
  "total_tail_gates_3": 8,
  "total_speeding_vehicles": 37,
  "total_standing_vehicles": 43,
  "total_standing_vehicles_shoulder": 0,
  "top_speed": 50.79204307919363,

```

```

    "average_velocity_north": 12.275952080722998,
    "average_velocity_south": 28.856717723740825,
    "traffic_jam_north": 0,
    "traffic_jam_south": 0,
    "slow_moving_traffic_north": 0,
    "slow_moving_traffic_south": 0,
    "total_trajectories": 301,
    "total_breakdowns_shoulder": 0,
    "total_breakdowns_driving_lane": 0,
    "total_breakdowns": 0,
    "total_accidents": 1
  }

```

**Listing 5.2:** Scenario statistics of the accident rosbag on the 08.04.2021

- Accident event on the 21.10.2021:

Listing 5.3 shows the JSON file containing the scenario statistics of the accident rosbag on the 21.10.2021, which was already depicted in Figure 4.3. While this event does not have any standing vehicles, traffic jams, slow-moving traffic, and breakdowns, it does have an accident. This is not being detected, since the rules proposed in Section 4.2.2.4 are only able to find accidents between two vehicles. However, in this case, the wind knocked over the blue van. Additionally, no standing vehicles are present meaning the current object detection is not able to detect the accident vehicle. Therefore, this case can not be detected by the proposed rule-based accident detection.

```

{
  "total_vehicles": 151,
  "total_vehicle_classes": 3,
  "total_lane_changes_left": 11,
  "total_lane_changes_right": 29,
  "total_lane_changes": 40,
  "max_lane_changes": 4,
  "total_cut_ins_left": 0,
  "total_cut_ins_right": 0,
  "total_cut_ins": 0,
  "total_cut_outs_left": 0,
  "total_cut_outs_right": 0,
  "total_cut_outs": 0,
  "total_tail_gates_1": 3,
  "total_tail_gates_2": 1,
  "total_tail_gates_3": 9,
  "total_speeding_vehicles": 35,
  "total_standing_vehicles": 0,
  "total_standing_vehicles_shoulder": 0,
  "top_speed": 49.008790563263396,
  "average_velocity_north": 31.39239954986089,
  "average_velocity_south": 21.478301163652848,
  "traffic_jam_north": 0,
  "traffic_jam_south": 0,
  "slow_moving_traffic_north": 0,
  "slow_moving_traffic_south": 0,
  "total_trajectories": 151,
  "total_breakdowns_shoulder": 0,
  "total_breakdowns_driving_lane": 0,
  "total_breakdowns": 0,

```

```

    "total_accidents": 0
  }

```

**Listing 5.3:** Scenario statistics of the accident rosbag on the 21.10.2021

- Accident event on the 28.03.2022:

Listing 5.4 depicts the JSON file for the accident on the 28.03.2022, where a black vehicle is driving into the side of a blue vehicle as shown in Figure 4.4. The two standing vehicles on a driving lane as well as the average velocity difference of 20 m/s between the northern and the southern side of the highway already hints at a potential accident in the recorded rosbag. A traffic jam or slow-moving traffic is not detected, since the slow-moving traffic is only occurring on a certain area of the A9 test stretch and not over the full length of the 500 m. Moreover, no breakdowns were present in the rosbag. However, similar to the accident event on the 21.10.2021 the occurred accident was not detected, because the developed rule-based approach can only detect rear-end collisions and not accidents, where the participants are driving in different lanes. Nonetheless, two standing vehicles were detected in a driving lane hinting at this possible accident.

```

{
  "total_vehicles": 3019,
  "total_vehicle_classes": 3,
  "total_lane_changes_left": 598,
  "total_lane_changes_right": 464,
  "total_lane_changes": 1062,
  "max_lane_changes": 10,
  "total_cut_ins_left": 5,
  "total_cut_ins_right": 13,
  "total_cut_ins": 18,
  "total_cut_outs_left": 5,
  "total_cut_outs_right": 5,
  "total_cut_outs": 10,
  "total_tail_gates_1": 1129,
  "total_tail_gates_2": 103,
  "total_tail_gates_3": 138,
  "total_speeding_vehicles": 265,
  "total_standing_vehicles": 2,
  "total_standing_vehicles_shoulder": 0,
  "top_speed": 50.89402821182269,
  "average_velocity_north": 10.254699696022493,
  "average_velocity_south": 30.311626097422366,
  "traffic_jam_north": 0,
  "traffic_jam_south": 0,
  "slow_moving_traffic_north": 0,
  "slow_moving_traffic_south": 0,
  "total_trajectories": 3019,
  "total_breakdowns_shoulder": 0,
  "total_breakdowns_driving_lane": 0,
  "total_breakdowns": 0,
  "total_accidents": 0
}

```

**Listing 5.4:** Scenario statistics of the accident rosbag on the 28.03.2022

- Breakdown event on the 22.05.2022:

Listing 5.5 illustrates the JSON file with the scenario statistics of the breakdown rosbag on the 22.05.2022, where a burning vehicle is standing on the shoulder lane as shown in figure 4.5. In this rosbag, a total of 5 vehicles are standing on the shoulder lane one of them being the burning vehicle ensuring the functional correctness of the standing shoulder label. Out of the 5 standing vehicles, 3 were standing continuously for 30s. Therefore, 3 breakdowns on the shoulder lane were detected ensuring the functional correctness of the breakdown. Additionally, slow-moving traffic on the southern side of the highway was recorded as well.

```
{
  "total_vehicles": 2663,
  "total_vehicle_classes": 3
  "total_lane_changes_left": 326,
  "total_lane_changes_right": 362,
  "total_lane_changes": 688,
  "max_lane_changes": 4,
  "total_cut_ins_left": 2,
  "total_cut_ins_right": 0,
  "total_cut_ins": 2,
  "total_cut_outs_left": 1,
  "total_cut_outs_right": 0,
  "total_cut_outs": 1,
  "total_tail_gates_1": 678,
  "total_tail_gates_2": 77,
  "total_tail_gates_3": 42,
  "total_speeding_vehicles": 230,
  "total_standing_vehicles": 5,
  "total_standing_vehicles_shoulder": 5,
  "top_speed": 51.28070425043302,
  "average_velocity_north": 27.593186592863535,
  "average_velocity_south": 15.574619147758387,
  "traffic_jam_north": 0,
  "traffic_jam_south": 0,
  "slow_moving_traffic_north": 0,
  "slow_moving_traffic_south": 1,
  "total_trajectories": 2663,
  "total_breakdowns_shoulder": 3,
  "total_breakdowns_driving_lane": 0,
  "total_breakdowns": 3,
  "total_accidents": 0
}
```

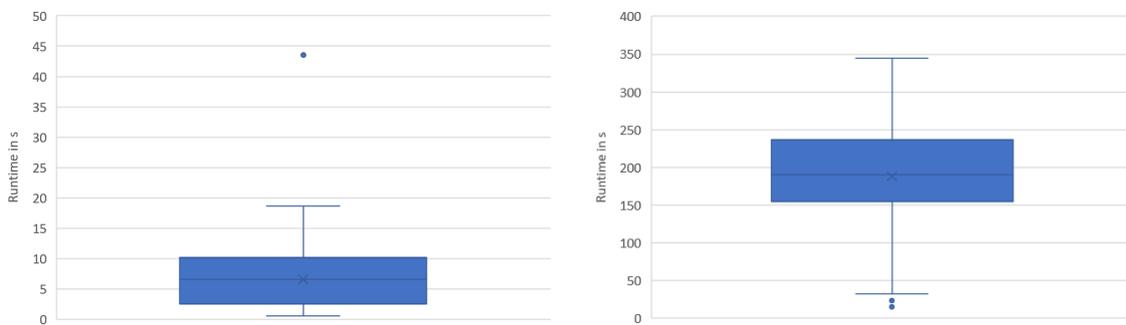
**Listing 5.5:** Scenario statistics of the accident rosbag on the 22.05.2022

All in all, this section presented the functional correctness of the standing shoulder label, the breakdown label, the slow-moving traffic, traffic jam as well as the accident label based on the data of the event log.

### 5.3.2 Automated Accident Detection

After presenting the real-world applicability of the scenario mining and the functional correctness of the maneuver detection, the automated accident detection is being evaluated. For the evaluation, the script was executed on the recorded rosbags from the A9 test stretch from the 04.12.2021 until the 30.04.2022, which consisted of 12,290 rosbags.

As already explained in Section 4.3, a standing script is being executed on the downloaded rosbag first to process the rosbags faster. Figure 5.2(a) shows a box plot, where the square inside the box represents the arithmetic mean, the horizontal line inside the box the median, the bottom line of the box the first quartile, which is the 25th percentile (this means, that 25% of the data lies below the line and the corresponding 75% above the line), the top line of the box the third quartile or 75th percentile, and the whiskers on the bottom and on the top visualize the 5th (bottom) and 95th (top) percentile [25] [13]. The data points below or above the whiskers are the outliers. Figure 5.2 shows the runtime for the standing detection as well as for the scenario detection, which were executed in the automated accident detection framework. Figure 5.2(a) shows the runtime the standing script needs for the rosbags. The runtime for the standing script varies between the 5th percentile at 0.55 s and the 95th percentile at 18.65 s with a median of 6.55 s and an arithmetic mean of 6.58 s. Additionally, an outlier is present at 43.57 s. Executing the standing script on all 12,290 rosbags takes 67,754 s, which is approximately 18.8 h.



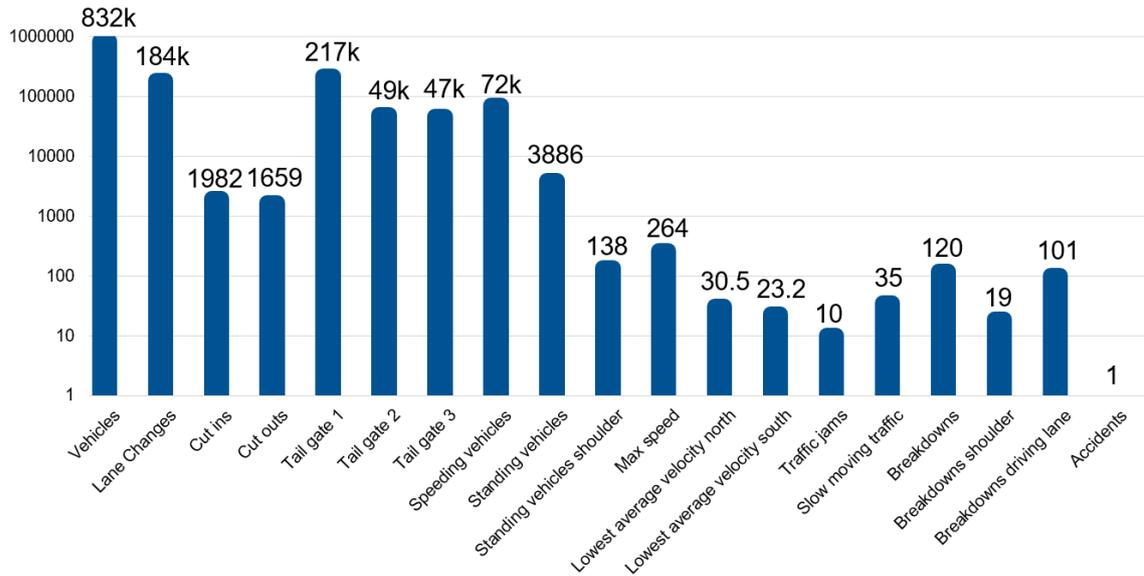
(a) Runtime needed for the script, which checks for standing vehicles in the rosbag. (b) Runtime needed for the scenario mining, which checks for any potential accidents in the rosbag.

**Figure 5.2:** Runtime needed for the computation of the standing vehicle detection and scenario detection for rosbags in the time interval of 5 months.

From the 12,290 executed rosbags, 336 rosbags contained standing vehicles, which corresponds to 2.73% meaning every 550 min a standing vehicle was detected. Therefore, the scenario mining was executed on those 336 rosbags individually. The runtime for the executed rosbags is shown in Figure 5.2(b), which ranges from the 5th percentile at 32.23 s to the 95th percentile at 345.20 s with a median of 190.67 s and an arithmetic mean of 187.72 s. Two outliers were recorded at 14.47 s and 22.66 s. The execution of the scenario mining for all 336 rosbags takes 63,073 s, which is roughly 17.52 h.

The scenario mining recorded the statistics shown in Figure 5.3 for the 336 executed rosbags. In those rosbags, a total amount of 831,969 unique vehicles were detected leading to 165 detected vehicles per minute. On average, every 4.5th vehicle changed the lane, since a total of 183,686 lane changes were recorded. According to Kaefer's statistics, 1,982 cut-ins and 1,659 cut-outs were detected. Additional important statistics from Kaefer are the tailgate events, where every fourth vehicle is responsible for a minor tailgate event, 5.9% were responsible for a moderate tailgate event and 5.6% vehicles were involved in a severe tailgate event. Out of the 831,969 recorded vehicles, a total of 72,000 are speeding, which comprise 8.7% and a total of 3,886 are standing summing up to 4.7%. Then, the standing vehicles can be split up again into standing vehicles on a driving lane, which are 3,748 or 96.4%, and standing vehicles on a shoulder lane, which are 138 or 3.6%. Additionally, the maximum recorded velocity was 73.45 m/s, which equals 264.42 km/h and is a newly recorded high

score on the A9 test stretch. Furthermore, the lowest recorded average velocity per rosbag on the northern side was 8.47 m/s and on the southern side 6.45m/s. Just as important, 120 breakdowns were recorded of which 19 occurred on a shoulder lane and 101 on a driving lane. Moreover, the scenario mining labeled 25 slow-moving traffic situations as well as 10 traffic jams in the 336 rosbags. While the A9 test stretch often records a traffic jam on the two highway exit lanes towards the south side, these were not recorded as traffic jams, since the vehicles on the other 3 lanes were driving fast enough that the average velocity was not below 20km/h. Last but not least, the rule-based accident detection found one accident in the recorded data.



**Figure 5.3:** Recorded scenario statistics on the 336 executed rosbags in the automated accident detection.

### 5.3.2.1 Precision

To evaluate the precision of the accidents, the extracted images were analyzed by hand. Out of the 26 detected accidents, 25 were false positives and 1 was a true positive. Therefore, the rule-based accident detection achieves a precision of 4% as shown in Equation 5.9.

$$precision = \frac{1}{1 + 25} = 0.04 \quad (5.9)$$

Figure 5.4 illustrates the time series of the true positive accident, which was detected due to the automated accident detection. On the bottom left of Figure 5.4(a), a vehicle is shown, that turned on the hazard warning lights and was coming close to a standstill on a driving lane due to a traffic jam forcing the traffic behind to slow down as well. However, in Figure 5.4(b) three vehicles behind this vehicle, a vehicle is being hit by the following vehicle whilst trying to change the lane, since it forgot to signal the lane change.

Figure 5.5 depicts different traffic situations, which the rule-based accident detection evaluated as an accident even though no accident occurred. Most of the false positive accidents are similar to Figure 5.5(a), Figure 5.5(c), and Figure 5.5(d), where a traffic jam occurs. Due to the close gap between vehicles, the YOLOv4 object detection is not able to



(a) Illustration of the traffic situation before the accident. (b) Illustration of the traffic situation during the accident.

**Figure 5.4:** Time series for the accident event on the 12.02.2022, which was detected due to the automated accident detection.



(a) Illustration of a false positive accident, where a traffic jam is recorded on the northern highway side.

(b) Illustration of a false positive accident, where construction workers are blocking multiple lanes on the southern highway side.



(c) Illustration of a false positive accident, where a traffic jam is recorded on the southern highway side.

(d) Illustration of a false positive accident, where a traffic jam is recorded on the southern highway side.

**Figure 5.5:** Illustration of multiple false positive accidents.

calculate the x and y position of the vehicles perfectly leading to bounding boxes that overlap. This often results in false positive accidents. Additionally, due to the limited training data, which was available to fine-tune the rules of the rule-based accident detection, such cases could not be filtered out. Other false positive cases are similar to the one in Figure 5.5(b), where construction workers are working on the highway and closing multiple lanes. This also leads to slow-moving traffic, which is hard to distinguish from accidents due to the used

YOLOv4 object detection.

### 5.3.2.2 False Negative Analysis

To detect additional accidents in the executed rosbags, the detected breakdowns were also analyzed, since they cover anomalous traffic behavior, which are likely to lead to accidents. With the help of the breakdown analysis, false negative accidents may be detected in the process. Out of the 120 breakdowns, 4 were false positives due to false positive detection from the YOLOv4 object detection and 116 true positives leading to a precision of 97% as shown in Equation 5.10:

$$precision = \frac{116}{116 + 4} = 0.97 \quad (5.10)$$

Figure 5.6 depicts four different traffic situations, which contain breakdown events. The first traffic situation is shown in Figure 5.6(a), where a police car is standing on the shoulder lane with blue lights. Secondly, Figure 5.6(b) depicts a standing vehicle on the leftest lane. Additionally, the accident on the 28.03.2022, which was shown in Figure 4.4, is visible on the right side of the picture. The third traffic situation is illustrated in Figure 5.6(c), where similar to the false positive accidents a traffic jam occurred, and lastly, Figure 5.6(d) shows a traffic situation, where construction work is being done. With the help of such anomalous traffic situations, potential interesting rosbags can be filtered out and analyzed by hand.



(a) Illustration of a true positive breakdown, where a police vehicle is standing on the shoulder lane.



(b) Illustration of a true positive breakdown, where a vehicle is standing on the shoulder lane.



(c) Illustration of a true positive breakdown, where a traffic jam occurred.



(d) Illustration of a true positive breakdown, where construction work is being done on a lane.

**Figure 5.6:** Illustration of multiple true positive breakdowns, which may lead to potential anomalous behavior or accidents.

### 5.3.3 Comparison with a Deep Learning-based Accident Detection

For the comparison of the rule-based and deep learning-based accident detection, a test set is created from the rosbags in the event log. To have a balanced test set with all important events, the following rosbags are being used:

- 11.02.2021 at 11:32: vehicle transporter present
- 08.04.2021 at 11:31: breakdown event on a driving lane right before the accident event
- 08.04.2021 at 11:32: accident event
- 15.05.2021 at 15:40: no event
- 15.05.2021 at 15:52: standing shoulder event
- 30.07.2021 at 09:25: traffic jam event
- 21.10.2021 at 10:21: accident event
- 28.03.2022 at 17:19: accident event
- 22.05.2022 at 17:14: breakdown event on the shoulder lane

#### 5.3.3.1 Precision

After creating the test set, the rule-based and deep learning-based accident detection was executed on all rosbags to calculate the precision of the accident detection. As already covered in Section 5.3.1.2, the proposed rule-based approach detects 1 out of the 3 accidents, since the goal was to detect the majority class of accidents, which are the detected rear-end collisions and then focus on finding new accident to train a deep learning-based accident detection. However, it does not falsely classify any traffic situations as accidents leading to 0 false positives. Therefore, the rule-based approach achieves a precision of 100% on the test set as shown in Equation 5.11. On the other hand, the deep learning-based accident detection can find all 3 out of 3 accidents and falsely classifies 1 traffic situation as an accident resulting in a precision of 75% as shown in Equation 5.12:

$$precision_{rb} = \frac{1}{1+0} = 1.00 \quad (5.11)$$

$$precision_{dl} = \frac{3}{3+1} = 0.75 \quad (5.12)$$

#### 5.3.3.2 Recall

In addition to the precision, both approaches were also compared using the recall on the provided test set. As already covered in Section 5.3.1.2, the proposed rule-based approach detects 1 out of the 3 accidents leading to a recall of 33% as shown in Equation 5.13. On the other hand, the deep learning-based accident detection can find all 3 out of 3 accidents resulting in a recall of 100% as shown in Equation 5.14:

$$recall_{rb} = \frac{1}{1+2} = 0.33 \quad (5.13)$$

$$recall_{dl} = \frac{3}{3+0} = 1.00 \quad (5.14)$$

### 5.3.3.3 Runtime

Additionally, the runtime of both approaches was compared, since the accident detection needs to analyze a large amount of recorded rosbags in as few time as possible. For the runtime evaluation, 4 different 1-minute rosbags from the event log were executed 10 times each:

- 08.04.2021 at 11:31: Rosbag comprising the sensor data of 4 different cameras
- 08.04.2021 at 11:32: Rosbag comprising the sensor data of 4 different cameras
- 11.05.2022 at 16:14: Rosbag comprising the sensor data of 2 different cameras
- 11.05.2022 at 16:29: Rosbag comprising the sensor data of 2 different cameras

Table 5.2 shows the runtime comparison for the four different 1-minute rosbags. The runtime of the rule-based accident detection fluctuates between 5.02 s and 8.63 s for the four rosbags because of the amount of vehicles in the rosbag, since the distance to leading and following vehicle scales exponentially with the amount of detected vehicles in a lane. However, the runtime of the deep learning-based approach needs for the same rosbags between 240.43 s and 484.69 s, because the images for the deep learning-based approach have to be extracted first and the model itself is more complex. Therefore, the average runtime for a rosbag with two cameras is 5.17 s for the rule-based approach and 244.30 s for the deep learning-based approach. For rosbags with four cameras, the runtime is then 7.61 s and 479.69 s respectively. Overall, the rule-based approach achieves an average runtime on all four rosbags of 6.39 s, while the deep learning-based approach takes 361.99 s. Therefore, the rule-based accident detection is 56.65 times faster than the deep learning-based approach.

Event	Runtime rule-based approach	Runtime deep learning-based approach
08.04.2021 at 11:31	8.63 s	484.69 s
08.04.2021 at 11:32	6.60 s	474.69 s
11.05.2022 at 16:14	5.02 s	240.43 s
11.05.2022 at 16:29	5.33 s	248.16 s
Average runtime for a rosbag with 2 cameras	5.17 s	244.30 s
Average runtime for a rosbag with 4 cameras	7.61 s	479.69 s
Overall average runtime	6.39 s	361.99 s

**Table 5.2:** Comparison of the runtime of the rule-based and the deep learning-based approach on four different 1-minute rosbags.

## 5.4 Discussion

With the results for the metrics presented in Section 5.1, the question if the scenario mining is performant enough to be used on multiple years of recordings and if the accident detection yields desirable results to create a training set containing anomalous traffic situations can be answered.

Firstly, the results for the scenario mining approach showed a significant runtime speedup of 18.80 compared to the previous work by Kaefer [1]. Additionally, the functional correctness

of the scenario mining was shown, since it can detect standing vehicles on the shoulder lane, slow-moving traffic, traffic jams, breakdowns, and most importantly accidents on the provided rosbags from the event log. However, it was also shown that the rule-based accident detection is limited to detecting crashes between two or more vehicles, which are on the same lane.

With a scalable and real-world applicable scenario mining, the contributions are extended of an automated accident detection, which was executed on 12,290 unseen rosbags from the A9 test stretch. During this 5-month recording interval, 831,969 vehicles, 120 breakdowns, 35 slow-moving traffic situations, 10 traffic jams, and 26 accidents were detected by the automated accident detection. In summary, the total runtime for the standing script and the scenario mining took roughly 36.32 h, which is feasible for the analysis of the recorded rosbags in the 5-month interval making the scenario mining applicable to analyze new data of multiple years. Additionally, the rule-based accident detection achieves a precision of 4%, which is compared to state-of-the-art approaches fairly low. Moreover, the reason for the low precision is the lack of training data to fine-tune the rules of the accident detection and the misclassification of the bounding boxes from the YOLOv4 object detection since most of the false positives occur in traffic jam scenarios. Additionally, due to the limited training data from the A9 test stretch, current state-of-the-art accident detections are not applicable to this use case causing the need for this simple first accident detection prototype to find more accidents, which can be used as training data for more complex models in the future. To detect false negatives of the rule-based accident detection, which are actual accidents that were not classified as an accident, the scenario mining and therefore also the automated accident detection extracted the images of all breakdown events. After analyzing the images by hand, 116 out of the 120 detected breakdowns were true positives resulting in a precision for the breakdown detection of 97%. While only one accident was found in the breakdown events by accident as shown in Figure 5.6, all breakdown rosbags contain anomalous traffic situations, which are useful for an anomaly traffic dataset. Additionally, deeper analyses of the origin of the breakdown events may lead to the finding of additional accidents. However, it is important to consider the possibility that the recorded data may not contain any additional accidents.

After evaluating the potential of the automated accident detection to find new accidents, it is being compared with a state-of-the-art deep learning-based accident detection written by Lehmborg in parallel to this work. The comparison was done using a subset of rosbags from the event log as test data. While the rule-based accident detection achieved a precision on the test data of 100%, the deep learning-based accident detection only achieved a precision of 75%. However, the deep learning-based accident detection has a better recall of 100% compared to the recall of 33% of the rule-based approach, because the deep learning-based accident detection can find all kinds of accidents since it is based on images, while the rule-based approach is only able to find rear-end collisions. Additionally, the runtime performance of both approaches was compared, where the rule-based accident detection achieves an average runtime of 6.39 s, while the deep learning-based accident detection needs on average 361.99 s for a single rosbag. Therefore, the proposed approach from Lehmborg is not suitable for the analysis of large amounts of recorded data making the rule-based accident detection the preferred approach to analyze the recorded data from the A9 test stretch.

Additionally, the two research questions have to be covered:

1. "Is it possible to reliably detect accidents on the A9 test stretch using roadside sensors?"

According to the previous definition of reliable in Section 1.2, the accident detection needs to achieve a precision of 0.7 and a recall of 0.85. While both approaches yield a

precision higher than 0.7, the rule-based accident detection only has a recall of 33%, while the deep learning-based accident detection has a recall of 100%. Therefore, only the deep learning-based accident detection can reliably detect accidents on the A9 test stretch using roadside sensors making it the preferred choice to analyze anomalous traffic situations.

2. "Is it possible to detect accidents on the A9 test stretch in real-time using roadside sensors?"

As shown in Table 5.2 the rule-based accident detection needs 6.39 s to analyze a 1-minute rosbag, while the deep learning-based accident detection needs on average 361.99s . Therefore, only the rule-based accident detection is capable of analyzing 25 FPS in real-time making it the preferred choice to go over a large dataset of unseen traffic data.

All in all, the proposed rule-based accident detection achieves a significant performance speedup compared to both Kaefer's scenario mining approach and Lehmborg's accident detection and can detect multiple anomalous events such as breakdowns, traffic jams, and accidents. Therefore, the proposed accident detection can be applied to detect accidents in the unseen data from the A9 test stretch.

# Chapter 6

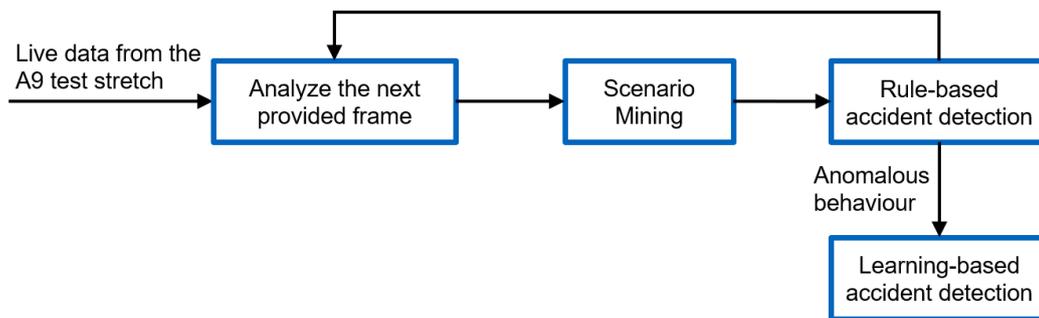
## Outlook

While the proposed automated accident detection together with the rule-based approach yielded promising results, further improvement can be achieved by extending it of the following features:

- As of now, the rule-based accident detection is only able to detect rear-end collisions. However, extending the approach for different accident scenarios helps in creating a larger anomaly traffic dataset. Therefore, instead of only calculating the distance between vehicles in a lane, also checking the distance between vehicles in neighbored lanes extends the Operational Design Domain (ODD) of the accident detection. Additionally, analyzing the trajectory of every vehicle can help detect accidents between a vehicle and an object such as traffic barriers, trees, utility poles or even accidents like the one on the 21.10.2021, where the vehicle was knocked over by the wind.
- The automated accident detection was only executed on a time interval of 5 months due to limited time, which only covers approximately 6.5% of the total 200.000 rosbags stored on the LRZ storage taking up 100 TB storage. Therefore, executing the automated accident detection on the additional unseen data could lead to multiple new accidents, with which the rule-based and the learning-based approach could be fine-tuned further. Additionally, analyzing and visualizing the accident found in this paper can help to improve both accident detections.
- The digital twin was created multiple years ago and used YOLOv4 for object detection. Executing the perception pipeline with a newer YOLO version to create a new digital twin with improved object detection and tracking can lead to better 3D position estimation and velocity estimation. This directly influences the accuracy of the scenario mining, since labels such as breakdowns and accidents depend on the continuous detection of vehicles as well as correct position estimation to make reliable predictions.
- Extending the digital twin of features such as police lights and hazard warning lights can help in detecting more anomalous traffic situations. For example, police lights are an indication of accidents and hazard warning lights can be used to detect breakdown events in a driving lane even during traffic jams, which can not be detected at the moment.
- Extend the image extraction of traffic jams in the scenario mining approach to further analyze such anomalous traffic situations, since accidents are one of the top 5 causes of traffic jams [34].
- Coupling the rule-based accident detection with the deep learning-based accident detection from Lehmborg can yield desirable results in the future. The idea is to first

execute the automated accident detection on the unseen recorded data and to extract images for anomalous events such as breakdowns, traffic jams and accidents. Afterward, the deep learning-based accident detection can be executed on the rosbags from the rule-based accident detection to make a final and reliable classification if an accident is present or not.

- Additionally, the rule-based and deep learning-based accident detection can be integrated into the live system as shown in Figure 6.1. The live system gets the data from the A9 test stretch as input and applies the scenario mining approach to each frame individually, which is responsible for calculating all features and maneuvers. Most importantly, it then calls the rule-based accident detection, which as shown in Table 5.2 is real-time capable. Therefore, after the scenario mining is finished and the rule-based accident detection annotated a frame as anomalous or not, the next frame is analyzed. Additionally, the rule-based accident detection calls the learning-based accident detection from Lehmborg if an anomalous behavior was detected to make a reliable prediction. The problem of the learning-based accident detection not being able to detect accidents in real-time is being solved by only executing it on the few anomalous frames. Therefore, by combining both accident detections the proposed approach can reliably detect accidents in real-time in the live system.



**Figure 6.1:** Integration of the rule-based and deep learning-based accident detection into the live system.

- After a thorough analysis of the highway recordings, the accident detection can also be applied to the recordings of intersections, since the project is also recording urban areas.

Therefore, with an improved automated accident detection, a larger anomaly traffic dataset can be created. With the help of such a dataset, Lehmborg's deep learning-based accident detection can be fine-tuned to outperform the rule-based approach in the long run leading to safer traffic behavior and faster emergency response time.

# Chapter 7

## Conclusion

This paper tackles the critical aspect of ensuring road safety and reducing the impact of road incidents by proposing an automated accident detection, which can be used for safer traffic behavior and faster emergency response time. Firstly, an event log was created from the anomalous rosbags, which were recorded on the A9 test stretch and already labeled manually. Secondly, the scenario mining approach from Kaefer [1] was used as baseline model for the rule-based accident detection. Due to Kaefer's slow feature extraction, the runtime of the lane ID and distance to leading and following vehicle extraction was improved. Afterward, the maneuver detection was extended of standing shoulder events, slow-moving traffic as well as traffic jams, breakdowns and most importantly accidents, which was trained on the rosbags provided by the event log. Thirdly, an automated accident detection was proposed, which downloads rosbags from a cloud storage, runs a standing vehicle classification and if standing vehicles are detected in the rosbag, runs the scenario mining approach with the rule-based accident detection. For all rosbags, on which the scenario mining was executed, images for the breakdown and accident events are extracted and the scenario statistics are generated to analyze the rosbags afterward and to create an anomaly traffic dataset from them. In the evaluation, the scenario mining approach achieved a speedup of 18.80 compared to Kaefer's approach [1]. Additionally, the functional correctness for all implemented maneuvers was shown with the help of the event log. Moreover, the automated accident detection was executed on 12,290 rosbags ranging over a time interval of 5 months. The script took roughly 36.32 h for the standing classification and the scenario mining approach achieving a precision of 4%, since only one accident was detected due to the automated accident detection. The detected false positives were due to the outdated YOLOv4 object detection and limited training data. Furthermore, a false negative analysis was done by examining all detected breakdowns. While multiple different anomalous traffic situations were found including an already known accident, no new accidents were established. Last but not least, the precision, recall and runtime of the rule-based accident detection was directly compared with a deep learning-based accident detection on a manually created test set from the event log. While the rule-based accident detection achieves a higher precision of 100% compared to the 75% of the deep learning-based approach, a lower recall of 33% compared to 100% was recorded, because the rule-based accident detection is limited to rear-end collisions, where both vehicles drive on the same lane. However, the rule-based approach achieves a significant runtime speedup of 56.65, since it only needs 6.39 s to process a single rosbag compared to the 361.99 s the deep learning-based approach takes on average making the rule-based accident detection more suitable for the analysis of large amounts of recorded data. Therefore, the rule-based accident detection is real-time capable, while the deep learning-based accident detection can detect accidents reliably making the combination of both approaches attractive for future projects. In conclusion, the exploration of this automated rule-based accident detection on the A9 test stretch using roadside sensors aids in creating a larger anomaly traffic dataset, which can be

used in the future to fine-tune a more sophisticated deep learning-based accident detection, which achieves a higher accuracy, to enhance the road safety and emergency response time.

# List of Abbreviations

**LiDAR** Light Detecting and Ranging

**radar** Radio Detection and Ranging

**FOV** Field of View

**FP** False Positive

**XML** Extensible Markup Language

**EDM** Euclidean Distance Matrix

**YOLO** You Only Look Once

**TTC** Time-to-collision

**JSON** JavaScript Object Notation

**ADAS** Advanced Driver Assistance Systems

**SSD** Solid State Drive

**CNN** Convolutional Neural Network

**RNN** Recurrent Neural Network

**ROS** Robot Operating System

**CCTV** Closed Circuit Television

**SORT** Simple Online Real-time Tracking

**SVM** Support Vector Machine

**R-CNN** Region-based Convolutional Neural Network

**MOG** Mixture of Gaussian

**MLP** Multilayer Perceptron

**CPU** Central Processing Unit

**LTS** Long Term Support

**RAM** Random Access Memory

**COCO** Common Objects in Context

**ODD** Operational Design Domain

**FPS** Frames Per Second

# List of Figures

- 1.1 Trend in the number of road traffic fatalities in the EU [6]. . . . . 1
  
- 4.1 Overview of the Event Log. . . . . 12
- 4.2 Time series for the accident event on the 08.04.2021, where a yellow vehicle crashed into a standing white van on the highway. . . . . 13
- 4.3 Time series for the breakdown on 21.10.2021, where a blue van was knocked over by the wind. . . . . 14
- 4.4 Time series for the accident event on the 28.03.2022, where a black vehicle is driving into the side of a blue vehicle and spinning it 360 degrees. . . . . 14
- 4.5 Time series for the breakdown event on the 22.05.2022, where a burning vehicle is standing on the shoulder lane. . . . . 15
- 4.6 Illustration of the different lane IDs on the highway. . . . . 16
- 4.7 Example of the lane ID labeling. . . . . 17
- 4.8 Example of the distance to leading and following vehicle calculation. . . . . 18
- 4.9 Illustration of the split of the average velocity per frame into four different categories. . . . . 19
  
- 5.1 Illustration of the folder structure of the image extraction. . . . . 30
- 5.2 Runtime needed for the computation of the standing vehicle detection and scenario detection for rosbags in the time interval of 5 months. . . . . 37
- 5.3 Recorded scenario statistics on the 336 executed rosbags in the automated accident detection. . . . . 38
- 5.4 Time series for the accident event on the 12.02.2022, which was detected due to the automated accident detection. . . . . 39
- 5.5 Illustration of multiple false positive accidents. . . . . 39
- 5.6 Illustration of multiple true positive breakdowns, which may lead to potential anomalous behavior or accidents. . . . . 40
  
- 6.1 Integration of the rule-based and deep learning-based accident detection into the live system. . . . . 46



# List of Tables

- 4.1 Overview of the statistics added to Kaefers [1] scenario statistics. . . . . 24
- 5.1 Average performance comparison of Kaefer’s approach with the approach proposed in this paper using the 15-minute rosbag on the 22.05.2022. . . . . 32
- 5.2 Comparison of the runtime of the rule-based and the deep learning-based approach on four different 1-minute rosbags. . . . . 42



## Bibliography

- [1] Aaron Kaefer Walter Zimmer, A. K. “Deep Traffic Scenario Mining, Detection, Classification and Generation on the Autonomous Driving Test Stretch using the CARLA Simulator”. In: *SUMO User Conference 2022* (2022), pp. 1–175.
- [2] ASAM. *ASAM OpenDRIVE*. URL: <https://www.asam.net/standards/detail/opendrive/> (visited on 01/08/2024).
- [3] *AUTOtech.agil*. URL: <https://www.mos.ed.tum.de/ftm/forschungsfelder/team-av-safe-operation/autotechagil/> (visited on 01/03/2024).
- [4] Bacon, J., Bejan, A. I., Beresford, A. R., Evans, D., Gibbens, R. J., and Moody, K. “Using Real-Time Road Traffic Data to Evaluate Congestion”. In: *Dependable and Historic Computing: Essays Dedicated to Brian Randell on the Occasion of His 75th Birthday*. Ed. by Jones, C. B. and Lloyd, J. L. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 93–117. ISBN: 978-3-642-24541-1. DOI: 10.1007/978-3-642-24541-1\_9. URL: [https://doi.org/10.1007/978-3-642-24541-1\\_9](https://doi.org/10.1007/978-3-642-24541-1_9).
- [5] Cheng, C.-H., Yang, J.-H., and Liu, P.-C. “Rule-based classifier based on accident frequency and three-stage dimensionality reduction for exploring the factors of road accident injuries”. In: *PLOS ONE* 17 (Aug. 2022), e0272956. DOI: 10.1371/journal.pone.0272956.
- [6] Commission, E. *Road safety statistics 2022 in more detail*. URL: [https://transport.ec.europa.eu/background/road-safety-statistics-2022-more-detail\\_en](https://transport.ec.europa.eu/background/road-safety-statistics-2022-more-detail_en) (visited on 01/15/2024).
- [7] Darwish, A. and Jain, A. “A rule based approach for visual pattern inspection”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 10.1 (1988), pp. 56–68. DOI: 10.1109/34.3867.
- [8] Doshi, K. and Yilmaz, Y. “Fast Unsupervised Anomaly Detection in Traffic Videos”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2020, pp. 2658–2664. DOI: 10.1109/CVPRW50498.2020.00320.
- [9] EvidentlyAI. “Accuracy vs. precision vs. recall in machine learning: what’s the difference?” In: (). URL: <https://www.evidentlyai.com/classification-metrics/accuracy-precision-recall> (visited on 01/27/2024).
- [10] Gelder, E. d., Manders, J., Grappiolo, C., Paardekooper, J.-P., Camp, O. O. d., and Schutter, B. D. “Real-World Scenario Mining for the Assessment of Automated Vehicles”. In: *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, Sept. 2020. DOI: 10.1109/itsc45102.2020.9294652. URL: <http://dx.doi.org/10.1109/ITSC45102.2020.9294652>.
- [11] Ghosh, S., Sunny, S. J., and Roney, R. “Accident Detection Using Convolutional Neural Networks”. In: *2019 International Conference on Data Science and Communication (IconDSC)*. 2019, pp. 1–6. DOI: 10.1109/IconDSC.2019.8816881.

- [12] Harlow, C. and Wang, Y. “Automated Accident Detection System”. In: *Transportation Research Record* 1746.1 (2001), pp. 90–93. DOI: 10.3141/1746-12. eprint: <https://doi.org/10.3141/1746-12>. URL: <https://doi.org/10.3141/1746-12>.
- [13] Herbein, S., Ahn, D., Lipari, D., Scogland, T., Stearman, M., Grondona, M., Garlick, J., Springmeyer, B., and Taufer, M. “Scalable I/O-Aware Job Scheduling for Burst Buffer Enabled HPC Clusters”. In: May 2016, pp. 69–80. DOI: 10.1145/2907294.2907316.
- [14] Hozhabr Pour, H., Li, F., Wegmeth, L., Trense, C., Doniec, R., Grzegorzec, M., and Wismüller, R. “A Machine Learning Framework for Automated Accident Detection Based on Multimodal Sensors in Cars”. In: *Sensors* 22.10 (2022). ISSN: 1424-8220. DOI: 10.3390/s22103634. URL: <https://www.mdpi.com/1424-8220/22/10/3634>.
- [15] Kamijo, S., Matsushita, Y., Ikeuchi, K., and Sakauchi, M. “Traffic monitoring and accident detection at intersections”. In: *IEEE Transactions on Intelligent Transportation Systems* 1.2 (2000), pp. 108–118. DOI: 10.1109/6979.880968.
- [16] Le, T.-N., Ono, S., Sugimoto, A., and Kawasaki, H. “Attention R-CNN for Accident Detection”. In: *2020 IEEE Intelligent Vehicles Symposium (IV)*. 2020, pp. 313–320. DOI: 10.1109/IV47402.2020.9304730.
- [17] LeCun, Y., Bengio, Y., and Hinton, G. “Deep learning”. In: *nature* 521.7553 (2015), pp. 436–444.
- [18] Li, Y., Wu, J., Bai, X., Yang, X., Tan, X., Li, G., Wen, S., Zhang, H., and Ding, E. “Multi-Granularity Tracking with Modularized Components for Unsupervised Vehicles Anomaly Detection”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2020, pp. 2501–2510. DOI: 10.1109/CVPRW50498.2020.00301.
- [19] Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., and Dollár, P. *Microsoft COCO: Common Objects in Context*. 2015. arXiv: 1405.0312 [cs.CV].
- [20] Liu, J., Martinez, L., Calzada, A., and Wang, H. “A novel belief rule base representation, generation and its inference methodology”. In: *Knowledge-Based Systems* 53 (2013), pp. 129–141. ISSN: 0950-7051. DOI: <https://doi.org/10.1016/j.knosys.2013.08.019>. URL: <https://www.sciencedirect.com/science/article/pii/S0950705113002554>.
- [21] Malik, M. B., Ghazi, M. A., and Ali, R. “Privacy Preserving Data Mining Techniques: Current Scenario and Future Prospects”. In: *2012 Third International Conference on Computer and Communication Technology*. 2012, pp. 26–32. DOI: 10.1109/ICCCT.2012.15.
- [22] Mehrannia, P., Bagi, S. S. G., Moshiri, B., and Al-Basir, O. A. “Deep Representation of Imbalanced Spatio-temporal Traffic Flow Data for Traffic Accident Detection”. In: *CoRR* abs/2108.09506 (2021). arXiv: 2108.09506. URL: <https://arxiv.org/abs/2108.09506>.
- [23] Organization, W. H. *Road traffic injuries*. URL: <https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries> (visited on 01/15/2024).
- [24] Raroque, C. *Differences Between Machine Learning, Artificial Intelligence, and Deep Learning*. URL: <https://aloo.co/blog/differences-between-machine-learning-artificial-intelligence-and-deep-learning> (visited on 02/05/2024).
- [25] Rebecca, S. “Box and whisker plot”. In: (). URL: [https://datavizcatalogue.com/methods/box\\_plot.html](https://datavizcatalogue.com/methods/box_plot.html) (visited on 01/26/2024).
- [26] Robotics, O. *ROS - Robot Operating System*. URL: <https://www.ros.org/> (visited on 01/18/2024).

- [27] S Pillai, M., Chaudhary, D.-G., Khari, M., and Gonzalez Crespo, R. “Real-Time Image Enhancement for an Automatic Automobile Accident Detection through CCTV using Deep Learning”. In: *Soft Computing In Press* (Sept. 2021), pp. 1–12. DOI: 10.1007/s00500-021-05576-w.
- [28] Sabry, K. and Emad, M. “Road Traffic Accidents Detection Based On Crash Estimation”. In: *2021 17th International Computer Engineering Conference (ICENCO)*. 2021, pp. 63–68. DOI: 10.1109/ICENCO49852.2021.9698968.
- [29] Shaalan, K. and Khaled. “Rule-based Approach in Arabic Natural Language Processing”. In: *the International Journal on Information and Communication Technologies (IJICT)* 3 (June 2010), pp. 11–.
- [30] Sherif, H. M., Shedid, M. A., and Senbel, S. A. “Real time traffic accident detection system using wireless sensor network”. In: *2014 6th International Conference of Soft Computing and Pattern Recognition (SoCPaR)*. 2014, pp. 59–64. DOI: 10.1109/SOCPAR.2014.7007982.
- [31] Sheu, J.-B. “A sequential detection approach to real-time freeway incident detection and characterization”. In: *European Journal of Operational Research* 157.2 (2004), pp. 471–485. ISSN: 0377-2217. DOI: [https://doi.org/10.1016/S0377-2217\(03\)00209-1](https://doi.org/10.1016/S0377-2217(03)00209-1). URL: <https://www.sciencedirect.com/science/article/pii/S0377221703002091>.
- [32] Tseng, S. *New ways to report driving incidents on Google Maps*. URL: <https://blog.google/products/maps/new-ways-report-driving-incidents-google-maps/> (visited on 10/17/2023).
- [33] Wang, T., Kim, S., Ji, W., Xie, E., Ge, C., Chen, J., Li, Z., and Luo, P. *DeepAccident: A Motion and Accident Prediction Benchmark for V2X Autonomous Driving*. 2023. arXiv: 2304.01168 [cs.CV].
- [34] *What causes traffic jams: 5 reasons for congestion*. URL: <https://www.financialexpress.com/auto/car-news/what-causes-traffic-jams-5-reasons-for-congestion/3003344/> (visited on 01/29/2024).
- [35] Ye, F.-F., Yang, L.-H., Wang, Y.-M., and Lu, H. “A data-driven rule-based system for China’s traffic accident prediction by considering the improvement of safety efficiency”. In: *Computers and Industrial Engineering* 176 (2023), p. 108924. ISSN: 0360-8352. DOI: <https://doi.org/10.1016/j.cie.2022.108924>. URL: <https://www.sciencedirect.com/science/article/pii/S0360835222009123>.
- [36] Zheng, M., Li, T., Zhu, R., Chen, J., Ma, Z., Tang, M., Cui, Z., and Wang, Z. “Traffic Accident’s Severity Prediction: A Deep-Learning Approach-Based CNN Network”. In: *IEEE Access PP* (Mar. 2019), pp. 1–1. DOI: 10.1109/ACCESS.2019.2903319.