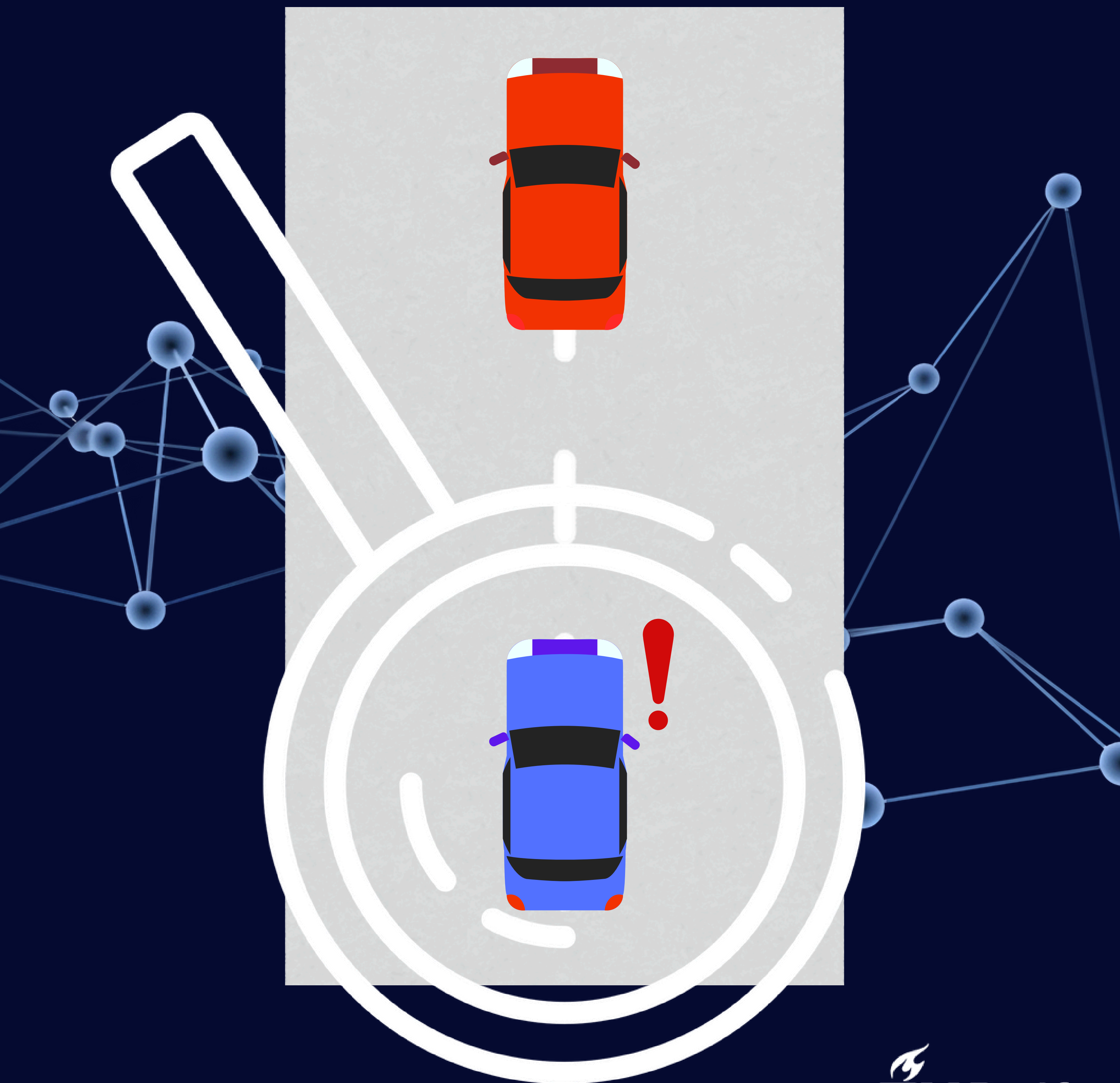
Detecting Long-term Behavioral Adaptations in Assisted Driving

An automated approach using neural networks and novelty detection



Rens Oude Elferink

TU Delft

Detecting Long-term Behavioral Adaptations in Assisted Driving

An automated approach using neural networks and novelty detection

by

Rens Oude Elferink

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Wednesday June 12, 2024.

Student number:	4708148	
Project duration:	September 26, 2023 – June 12, 2024	
Thesis committee:	Prof. dr. ir. L. C. Siebert,	TU Delft, supervisor
	Prof. dr. A. Lukina,	TU Delft, supervisor
	Prof. dr. C. A. Raman,	TU Delft

TU Delft

Preface

In today's world, as technology becomes more integrated into our lives, it is crucial to understand how humans and technology can effectively work together. This is especially relevant in safety-critical activities such as driving, where we depend on advanced systems for assistance. For these systems to function optimally, they must align with our preferences and behaviors. Failure to achieve this alignment can cause safety concerns and undermine our trust in them. This thesis explores methods using neural networks to identify and understand behavioral adaptations made to driving assistance systems. My goal is to further explore the interaction between drivers and vehicles while also improving our understanding of how neural networks operate.

My interest in this subject comes from two main sources. Firstly, I am fascinated by the idea of making Artificial Intelligence more understandable to humans. I believe that when people can grasp how neural networks work, it builds trust and makes collaboration between humans and technology easier. Secondly, I am passionate about exploring topics that bring together different areas of study. The combination of research on driving behavior with methods such as neural networks and novelty detection opens up exciting possibilities. This thesis is my exploration of this interdisciplinary field, aiming to discover new insights that benefit both disciplines. I am proud to say that through this thesis, I have been able to find a relevant direction to integrate neural networks and novelty detection into ongoing research on driving behavior.

I want to express my gratitude to my supervisors, Anna Lukina and Luciano Siebert, for their guidance throughout my thesis and for facilitating the collaboration between the departments of Algorithmics and Interactive Intelligence. This collaboration was particularly interesting, as it combined relevant theory with an interesting real-world application.

Rens Oude Elferink
Delft, June 2024

Summary

The autonomous vehicle industry has the potential to reshape the future of driving. As we move towards fully autonomous vehicles, understanding the interaction between vehicles and drivers is crucial. Advanced Driver Assistance Systems (ADAS) play a pivotal role in this transition, where these systems form the bridge between automation and manual driving. However, research suggests that humans adapt their driving behavior when using ADAS, which can affect safety. Although short-term behavioral effects have been studied, there is a significant gap in understanding long-term adaptations. Addressing this gap is essential for improving safety and optimizing the performance of these systems. This thesis explores a method to automatically detect these long-term behavioral adaptations, using a neural network combined with a technique for novelty detection.

The presented approach uses a neural network aimed at driver identification to detect behavioral adaptations on an individual level, while the novelty detection technique classifies this behavior as normal or abnormal. Key design considerations for effective detection of behavioral adaptations are discussed, including the need for a dataset that comprehensively represents normal behavior. For the neural network, essential factors include selecting informative input features, designing an architecture capable of extracting high-level driving information, and ensuring proper training for accurate detection. In addition, the novelty detection component must be able to accurately capture the normal behavior of the driver while also being able to indicate instances that deviate from this behavior. The final configuration consists of a neural network with multiple Long Short-Term Memory (LSTM) layers, with a novelty monitor that can effectively detect inputs not seen during training.

The approach is evaluated by using the CARLA simulator, combined with the Intelligent Driver Model to simulate and model a range of driving behaviors and adaptations. The experiments indicate that our approach is effective in detecting a behavioral adaptation of the driver, although the approach could be improved to better capture small adaptations in the driver's behavior. The results also indicate that the chosen approach to novelty detection is an effective method to indicate abnormal behaviors, where the experiments showed an intuitive high-level interpretation of the behavioral adaptation.

The main limitation of the method is its lack of realism. To improve it, the approach should be tested in more realistic environments that include changing road and weather conditions, as well as noisy sensors. Extending this approach is challenging, particularly in defining normal behavior in real-life situations, where 'normal' behavior can be a subjective concept. Future work could also focus on techniques for automatically setting the relevant parameters for novelty detection, which would eventually enable the approach to be used in real-time applications.

Contents

Nomenclature

Abbreviations

Abbreviation/Acronym	Meaning
ADAS	Advanced Driver Assistance Systems
BA	Behavioral Adaptation
FOT	Field Operational Test
IDM	Intelligent Driver Model
RNN	Replicator Neural Network
LSTM	Long Short-Term Memory
GMM	Gaussian Mixture Model
CNN	Convolutional Neural Network
HMM	Hidden Markov Model
SVM	Support Vector Machine

1 Introduction

The autonomous vehicle industry is experiencing rapid growth, underscored by the recent approval of the first conditionally autonomous vehicle on public roads in the United States by Mercedes-Benz [1]. Conditional autonomy implies that the driver is relieved from constant monitoring during car engagement but must remain responsive to intervention requests. In the future, the introduction of fully autonomous cars is expected, where the autonomous vehicle should be able to control all driving scenarios unconditionally [2]. Until automated vehicles can handle all traffic scenarios independently, understanding and improving the interaction between vehicle and driver is crucial to advancing driving technology safely.

Advanced Driver Assistance Systems (ADAS) form an important step in the transition phase toward autonomous driving. In recent years, significant progress has been seen in ADAS technology, indicated by the widespread adoption of various ADAS features in modern vehicles, including adaptive cruise control, collision avoidance systems, intelligent speed adaption, and parking assistance systems [3]. As a result, the efficiency and safety of driving have increased, leading to a decrease in the number of accidents [4]. With ADAS as a stepping stone, the automotive industry is aiming towards the realization of fully autonomous vehicles, potentially reshaping the future of transportation.

However, in the context of human-automation interaction, research has indicated that humans adapt their driving behavior in response to automation [5]. These behavioral adaptations (BA) have notable effects on the safety and effectiveness of ADAS. Research has revealed that drivers can, for example, show increased speeds when using features such as adaptive cruise control [6] or haptic steering systems [7]. Furthermore, behavioral adaptations can manifest in heightened engagement in secondary tasks [8] and slower reaction times to sudden events [9]. Such behavioral changes not only affect driving safety, but also contribute to the culpability gap [10]. In these cases, legal liability is often unfairly placed on drivers for crashes involving partially automated vehicles, despite the unrealistic expectation that they should remain vigilant and quickly take over control of automation. The tendency to adapt driving behavior within the context of ADAS usage raises clear concerns about vehicle-driver interaction and road safety. These concerns encourage a detailed exploration of how people adjust their behavior and the dynamics at play when advanced assistance is introduced in vehicles.

Past research investigating behavioral adaptations in driving has focused primarily on short-term effects [11]. Although this focus has provided valuable insights into immediate adaptations made by individuals who interact with driving assistance systems, it is believed that behavioral adaptations may manifest after longer periods, as drivers become more familiar with the assistance system at play [12]. However, only a limited number of studies have made approaches to indicate long-term adaptations, e.g. [13–17], even though the importance of long-term research has been frequently emphasized [5, 11, 18, 19]. This discrepancy indicates a gap in the research on behavioral adaptations. Addressing this gap is crucial for understanding drivers' behavioral changes over time with assistance systems, which is essential for improving safety and ensuring that these systems function as intended.

The detection of long-term behavioral adaptations in response to ADAS presents a significant challenge, which likely contributes to the limited number of studies in this area. Unlike immediate adaptations, which may be more easily identifiable using test track or simulator studies, the exact timing and nature of long-term adaptations remain elusive due to the costly process of collecting and analyzing the large amount of data resulting from long-term driving experiments [20].

However, with the rise of advanced analysis techniques, there is significant potential to develop methods capable of optimizing this research process. Especially neural networks and novelty detection offer promising solutions for identifying long-term behavioral adaptations. Neural networks are effective tools for pattern recognition and can manage large datasets effectively, making them suitable for analyzing the vast amounts of data generated by long-term driving studies. Novelty detection, a technique used to identify new or unusual patterns within a dataset, is particularly relevant in this context. In our problem setting, novelty refers to changes in driving behavior that deviate from normal behavior. The method in this thesis applies a neural network to detect changes in driving behavior on an individual level, while novelty detection is used to classify the change as normal or abnormal. This classification is crucial, since research often focuses on negative behavioral adaptations, e.g. [6–9]. Detecting abnormal behavior can thus serve as an indicator of such negative adaptations.

An effective and promising method that combines neural networks and novelty detection focuses on monitoring neural networks by observing hidden layers and creating abstractions that define the known inputs of the network [21, 22]. For this thesis, this method is particularly effective because it operates independently of prior knowledge about the nature of novelties. This is especially valuable given the unknown characteristics of long-term behavioral adaptations. The method creates abstractions that encompass the outputs of the neural network that define normal driving behavior. During runtime, any input not contained within these abstractions is detected as a novelty, allowing us to classify it as abnormal driving behavior. Another aspect that makes the method effective is that the method can be applied to any network architecture. The method has previously been shown to be effective in detecting novelties in image classification tasks using Convolutional Neural Networks [21]. This thesis extends the application of this method to the detection of novelties in time series using a Recurrent Neural Network.

Integrating advanced analysis techniques with current research methods, such as Field Operational Tests (FOTs), is the key to maximizing the impact of these techniques on long-term research. FOTs are large-scale experiments in which assistance systems are evaluated under normal driving conditions, typically lasting at least a few weeks [23]. As the integration of new sensors into vehicles becomes more widespread, the use of these sensors to collect data in FOTs becomes more cost-effective. However, these new sensors also cause an increase in collected data, which requires new and improved methods to handle and analyze these data [24]. Especially techniques capable of indicating relevant events within large-scale dynamic data streams, along with methods that uncover patterns that are often overlooked by traditional hypothesis testing, are particularly valuable [20]. Combining FOTs with advanced analysis techniques thus provides great potential for improving long-term research on behavioral adaptations.

This thesis aims to apply advanced analysis techniques, specifically a neural network and the abstract-based approach to novelty detection, to create a detector that can indicate relevant events containing behavioral adaptations. The research presented in this thesis is guided by the following research question: *How can long-term behavioral adaptations, as an effect of driving with assistance systems, be effectively detected using the abstract-based approach to novelty detection?* The subquestions created to break down the research are the following:

1. How can the detection process be implemented to ensure that detected events consistently correspond to instances of behavioral adaptations?

2. To what extent do the events identified by the detector correspond to behavioral adaptations of the driver?

The scope of this thesis is limited to car-following scenarios on highways, intentionally disregarding external factors such as weather conditions and changing road environments. Car-following was selected as the main scenario due to its prevalence in real-life driving situations [25, 26]. Moreover, this thesis exclusively considers performance changes of the driver, as opposed to other types of behavioral adaptation, such as cognitive changes or perceptive changes [3]. These performance changes are exemplified by adaptations in accelerations, velocity, or longitudinal and lateral deviations. This choice was made because indicators of these performance changes can be collected using built-in vehicle sensors, and limiting the scope to behavioral changes detectable through existing sensors allows for straightforward dataset collection and seamless integration of the detection system into FOTs.

In this thesis, we first introduce the necessary background in Chapter 2 to provide a solid foundation for understanding our work. Chapter 3 then offers an overview of other research related to our approach. Our methodology, detailed in Chapter 4, outlines the approach we adopt to address the first research question. In Chapter 5, we present the experimental setup used to validate our methodology and answer the second research question. This is followed by an analysis of the results in Chapter 6. The discussion of our thesis is given in Chapter 7, and finally, Chapter 8 offers a conclusive summary of our contributions.

2
Preliminary

This chapter provides the background necessary to understand the contributions in this thesis. The background for the nature and timing of behavioral adaptations is given in section 2.1. Then, section 2.2 introduces the Intelligent Driver Model, which is the model used in the thesis to simulate driving behavior and behavioral adaptations. Finally, section 2.3 describes the abstract-based approach to novelty detection, which is the technique used in this thesis to detect abnormal driving behaviors.

2.1. Behavioral adaptations

In the context of driving, the term behavioral adaptation (BA) describes a broad spectrum of adjustments that drivers make when introduced to changes in the vehicle. The Organisation for Economic Co-operation and Development (OECD) defines behavioral adaptation as *"Those behaviors which may occur following the introduction of changes to the road-vehicle-user system and which were not intended by the initiators of the change"* [27]. In this thesis, the focus is primarily on the negative (unsafe) adaptations made by drivers; however, the detector should also be capable of detecting more general adaptations.

The various forms of behavioral adaptation resulting from the integration of ADAS can be categorized into different groups, each containing different aspects of driver response. These groups are perceptive changes, cognitive changes, performance changes, changes in driver status, attitudinal changes, and changes in adaptation to environmental conditions [3]. Although some of these types of behavioral changes require specialized sensors for monitoring, such as those measuring attention or stress levels, others can be effectively captured using sensors already available in most modern vehicles, such as speed and forward distance sensors. In this study, the focus is on performance changes, which are contained in the latter of these two groups. These performance changes are all different adaptations that have a direct effect on driving parameters, such as the choice of speed, the magnitude of acceleration, the amount of braking, and the longitudinal and lateral deviations.

Behavioral adaptation tends to occur only after a learning phase has passed, in which the driver learns to drive effectively with the ADAS. This is followed by an integration phase, in which the driver is experienced enough to integrate the system into their driving [11]. It is possible that due to this temporal nature of BA, the positive effects of the use of ADAS revealed in short-term studies might not consist after long-term exposure [3]. The learning phase can be subdivided into the first encounter, learning, trust, adjustment, and readjustment phases [28]. Relative stability in behavior is generally attained during the trust phase, which is thought to last up to six months following the introduction of the assistance system. In subsequent phases, the driver may start to adapt their behavior to the assistance systems. Therefore, long-term research on behavioral adaptations should be conducted only after the trust phase has finished, that is, after six months.

2.2. Intelligent Driver Model

This thesis uses a driving simulator to generate the driving data. To effectively model various driving behaviors in car-following situations, we need an adaptable model. Among the range of available models for car-following, the Intelligent Driver Model (IDM) [29] is the most widely applied [30], providing a straightforward approach to simulate car-following behaviors on the road. Moreover, by adjusting its parameters, we can model different driving styles and behavioral adaptations. The IDM is defined using two formulas that set the acceleration of the following vehicle.

$$\frac{dv}{dt} = a\left(1 - \left(\frac{v}{v_0}\right)^{\delta} - \left(\frac{s^*}{s}\right)^2\right) \quad (2.1) \qquad s^* = s_0 + \max\left(0, v \cdot T + \frac{v \cdot \Delta v}{2\sqrt{ab}}\right) \quad (2.2)$$

Equation 2.1 shows the calculation of the acceleration, which is a function of the current velocity v, the desired velocity v_0, the acceleration exponent δ, the current gap s, and the desired minimum gap s^*. The desired minimum gap is defined by Equation 2.2, additionally using the minimum desired gap at standstill s_0, the desired time headway T, the difference in velocities of the vehicles Δv, the maximum acceleration of the vehicle a and the comfortable deceleration b. The main parameters of the IDM that define the behavior of the vehicle are v_0, a, b, δ, s_0, and T.

2.3. Abstract-based novelty detection

The novelty detection method used in this thesis is adapted from [21, 22]. This method offers a way to monitor neural networks and detect unseen inputs during training as novel during runtime. The approach assumes that the informative patterns present in the input data can be captured by extracting output from hidden layers near the end of the neural network. Moreover, they assume that these outputs accurately represent the class to which the input belongs.

By monitoring a selected hidden layer at the end of a neural network and extracting the output of this layer, it is possible to find the outputs of the hidden layer that correspond to each class. These outputs are then used for the creation of abstractions, in this case bounding boxes, that enclose the outputs corresponding to one specific class. During training, only correctly classified inputs are used for the creation of abstractions. The number of neurons in the hidden layer corresponds to the dimensions of the created bounding box. The process of creating such an abstraction is shown in Figure 2.1.

To apply this technique, the neural network under consideration must be pretrained. Generally, the training data of the neural network is also used to create the box abstractions. During runtime, any input of the network falling outside of these boxes is then considered to be novel. In this way, the novelty detection technique can detect inputs that deviate from the inputs seen during training. The high-level detection process is shown in Figure 2.2.

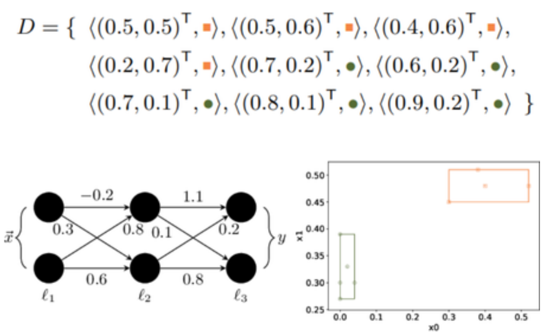


Figure 2.1: The abstraction creation process of the novelty detection technique. The example input is processed by the network and the outputs of layer l_2 are extracted. The graph shows the distribution of these points, indicating that the outputs are similar for similar classes. The abstraction encloses the area that corresponds to each class [21].

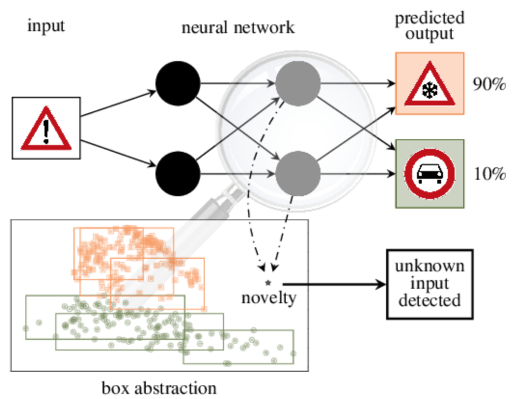


Figure 2.2: The high-level overview of the novelty detection technique [21]. Any input that has not been seen during training will not be contained in any of the abstractions, and is detected as novelty.

Figure 2.3: The abstraction creation process and high-level overview of the novelty detection technique.

This approach has two particularly useful advantages: it is compatible with various network architectures and does not require prior knowledge of the data distribution. These advantages ensure that the technique can be applied to a broad range of applications. Furthermore, novelty detection is efficient and does not require extensive computational resources.

3

Related work

This chapter describes previous approaches for detecting behavioral adaptations in a long-term setting (section 3.1) and the techniques that have been used throughout the literature to detect abnormal driving behaviors (section 3.2), identify drivers (section 3.3) and indicate novelties in time series (section 3.4).

3.1. Long-term behavioral adaptations

Only a limited number of approaches have focused on identifying behavioral adaptations in a long-term setting. These approaches differ in the type of behavior change and the setup they consider.

An approach that also considers the performance changes of the driver focuses on observing how a driver's behavior evolves over time as they become increasingly familiar with an autonomous vehicle [13]. They find that with repeated usage there is a positive change found in perceptive, attitudinal, cognitive, and driver state changes. Furthermore, this paper suggests that performance changes, indicated by takeover operations, may not become more pronounced with repeated usage. However, this approach focuses only on autonomous vehicles, rather than on assistance systems. In addition, the approach lacks consideration for an extended observation period necessary to detect long-term behavioral adaptations. Therefore, they cannot draw grounded conclusions regarding the long-term perspective.

An approach that considers long-term attitudinal changes, e.g. trust and acceptance, rather than performance changes, evaluates the behavior of drivers as they are introduced to an adaptive collision avoidance system [14]. In an evaluation phase of one month, drivers interact with the system through a simulator and are exposed to several dangerous lane-change maneuvers. The paper shows that the performance of the drivers improves over time, as the acceptance and understanding of the system increases. However, according to the literature, long-term is considered to be after 6 months, which makes it questionable whether this approach can be classified as long-term.

A more reasonable time period is considered by [15], where they consider behavioral adaptations to an intelligent speed adaptation system over a period of 6 months. They indicate more overriding behavior of the system as time progresses. They only track the number of overrides and do not focus on any other behavioral adaptations. In contrast, the approach presented in this thesis focuses more on changes in driving behavior, such as changes in acceleration and speed. Furthermore, they indicate that it is difficult to predict long-term adaptations using short-term experiment results, since they found that adaptations largely depend on circumstantial influences.

These circumstantial influences are best captured by FOTs, where more realistic environments are considered. Until now, there have been no FOTs with a specific focus on the performance changes of the driver. The current focus of FOTs has mainly been on the validation of current assistance systems. For example, a five-month FOT [31] demonstrated significant positive safety improvements with lane assistance systems and adaptive cruise control. These were exemplified by a decrease in critically short headway and fewer unintentional lane crossings. However, further analysis of this FOT [17] found that adaptive cruise control is frequently deactivated in dense traffic conditions, indicating a behavioral adaptation of drivers. Another FOT in China [32], although considering a shorter time period, showed positive results with forward collision sensors and lane departure warning systems, including increased braking time and decreased relative speed. Furthermore, they found that there was a significant influence of road type and experience on driving behaviors.

These approaches indicate that there is a lack of long-term research on behavioral adaptations, with most approaches failing to consider relevant extended time periods or focusing solely on specific aspects of behavior change. This limitation hinders the ability to draw comprehensive conclusions about long-term adaptations.

3.2. Anomalous driving behavior detection

To the best of our knowledge, there have been no approaches aimed at automatically detecting instances of behavioral adaptation in driving in the existing literature. However, there have been approaches that aim to solve a similar goal, which is the detection of anomalous driving behavior. This goal is similar to detecting behavioral adaptations, but while anomaly detection in driving mainly focuses on detecting sudden, unexpected events like fast braking or rapid swerving, behavioral adaptations focus on a high-level behavior drift over an extended period.

In the context of detecting anomalous behavior in driving, there have been multiple attempts that focus both on detecting anomalous events and detecting anomalous behavior. An example of the first case is the detection of events such as rapid acceleration, sudden braking, or rapid swerving [33]. They take an online detection approach to indicate driving anomalies. Similarly to this thesis, they do not rely on labeled data since this is rarely available for large datasets of driving data. The approach they take uses a state graph to create a behavior model, which allows them to accurately identify anomalous events. Another example is the detection of swerving, fast U-turns, or weaving using smartphone sensors [34]. Approaches that consider in-vehicle sensors have focused, for example, on detecting driver distraction [35], unusual driving behaviors [36], risky driving behaviors [37], and driving events [38]. These studies demonstrate that leveraging solely in-vehicle sensor data offers many opportunities to detect specific behaviors. However, these approaches do not take into account the behavior of an individual over an extended period and, as such, cannot be used to indicate behavioral adaptations.

Other approaches have the potential to detect these changes [39, 40] by considering longer periods and being able to indicate changes on a higher level. [39] applies a neural network for driver identification, incorporating a preprocessing step where they detect deviations from the anticipated driving behavior of the individual. Upon detecting such deviations, the neural network further verifies whether the driving data aligns with the actual driver of the vehicle. Our methodology shares similarities with this technique in employing a neural network to detect unexpected driving behavior of the individual. Nevertheless, the distinction lies in our application of anomaly detection to identify behaviors outside predefined normal classes, whereas the aforementioned approach applies anomaly detection to indicate the need for driver identification. This approach could be adapted to focus on behavioral adaptation detection, but their focus is on the identification of drivers for security reasons. In addition, they do not focus on the classification of abnormal driving behavior, which is focused on in this thesis.

The second paper showing potential for behavior change detection uses a Replicator Neural Network (RNN), Long Short-Term Memory (LSTM), and Isolation Forests to spot behavior different from normal driving [40]. They define normal and aggressive driving using a driver model for cars in a simulation. The RNN is trained to replicate driving data of normal behavior. They find that this RNN has the best performance for detecting aggressive driving, where these behaviors produce significantly higher reconstruction errors than normal driving. This method could potentially detect any unusual behavior, not just aggression. In their current setup, they currently do not indicate changes in behavior on an individual level. Currently, they detect general changes to predefined normal behavior. This method could be adapted to identify individual adaptations by training the RNN only on driving data from an individual. A high reconstruction error would then indicate a shift in individual behavior. In this thesis, our method is specifically designed to detect changes on an individual level. An overview of all the papers discussed in this section is given in Table 3.1.

3.3. Driver identification

This thesis aims to apply a neural network to indicate deviations from the expected driving behavior of an individual. An application of neural networks that shares a similar goal is that of driver identification, where driving behavior is matched to the corresponding driver. This method has potential since it can

Table 3.1: Overview of related work regarding anomalous driving behavior detection.

Paper	Input Data	Focus	Detection
[33]	In-vehicle sensor data	Detection of anomalous events	Detects sudden events, not long-term behavior changes
[34]	Smartphone sensor data	Detection of anomalous events	Detects sudden events, not long-term behavior changes
[35]	In-vehicle sensor data	Detection of driver distraction	Detects specific behaviors, not long-term behavior changes
[36]	Smartphone sensor data	Detection of anomalous events	Detects sudden events, not long-term behavior changes
[37]	In-vehicle sensor data	Detection of risky driving behaviors	Detects specific behaviors, not long-term behavior changes
[38]	In-vehicle sensor data	Detection of driving events	Detects specific behaviors, not long-term behavior changes
[39]	In-vehicle sensor data	Driver identification	Could detect behavior drift but focuses on driver identification
[40]	In-vehicle sensor data	Detection of abnormal (aggressive) driving behavior	Could detect unusual behavior, not currently focused on individual changes

also be applied to indicate when the behavior does not match with the corresponding driver, which can indicate a deviation from expected driving behavior.

Several approaches have explored the identification of drivers with neural networks [35, 41, 42], which has been shown to be an effective method for driver identification. The papers have a similar goal, but use different sample sizes and feature sets for this identification. For example, in [42], an extensive and comprehensive set of features is used, including acceleration, distance from the leading vehicle, lane information, speeds, and pedal operations. This approach achieves an identification rate of 83.1% for a larger group of 51 drivers. Other approaches have shown that a small feature set can already achieve accurate identification. This has been done using only vehicle acceleration data as primary input [41], resulting in an identification rate of 88.3% across a group of 13 drivers. Furthermore, a paper that focuses specifically on brake and gas pedal operations, attains an identification rate of 84.6% for a smaller sample size of three drivers [35]. These studies collectively indicate the efficiency of neural networks in identifying individual drivers based on various driving behaviors and parameters. This thesis aims to identify only one specific driver from a broader set of driving behaviors, rather than attempting to identify each individual driver. Therefore, the features used in these papers are expected to achieve a higher identification rate in our setting.

In the literature, multiple attempts have been made to effectively identify drivers in car-following situations. Although these attempts are not focused on neural networks, they are relevant, as they explore a car-following context similar to our approach. A paper that does this using vehicle sensors obtains an identification rate of 73% for 30 vehicles in an actual city area [43]. Using a Gaussian Mixture Model (GMM), they find that velocity, headway distance, and pedal operations are the signals that best define a specific driver. A similar approach [44] uses the same set of features and also applies a GMM to obtain an identification percentage of 76.8% over 267 drivers. Apart from GMMs, an approach using decision trees [45] was able to achieve an identification percentage of 82%. The set of features they used consisted of gas pedal operations, steering wheel angle, vehicle speed, RPM, and yaw rate. These approaches indicate the relevant features that have been used for driver identification in car-following situations.

3.4. Time series novelty monitoring

The novelty detection approach in this thesis aims to detect behavior that deviates from typical driving behavior, represented as a multivariate time series. To solve the problem of novelty detection in time series, multiple approaches have been made. Each of these approaches deviates in some way from the abstract-based approach we consider in this thesis.

A widespread technique in this field is the use of RNNs [46, 47]. These RNNs are designed to replicate the input by compressing and decompressing the data. The difference between the actual input and the reconstructed input is referred to as the reconstruction error. An example of such an approach describes the use of RNNs to detect shifts in the distribution of normal behavior [47]. Their approach involves gathering reconstruction errors over time, stating that sustained significant errors signify a behavioral shift, whereas sudden spikes in errors followed by a return to normal levels suggest an anomaly. RNNs could be a useful technique for identifying changes in behavior by training them on the normal driving behavior of an individual and then collecting reconstruction errors as the driver is exposed to ADAS. The difference from the abstract-based approach is that while the RNN presents a standalone application to novelty detection in time series, the abstract-based approach offers a flexible and integrated method to detect novel inputs to any neural network in real-time. This allows us to combine novelty detection to any neural network application, such as driver identification.

Other approaches use Hidden Markov Models (HMMs) to indicate novelties in time series [48, 49]. This is an interesting technique, but creating a good Markovian model for driving behavior is not straightforward. Moreover, the abstract-based novelty detection technique offers a more direct application to neural networks.

One-class Support Vector Machines (SVMs) are also used throughout the literature to indicate novelties in time series [50, 51]. However, the issue with these SVMs is that they require high computation, which means that their application in real-time applications is limited. In the case considered in this thesis, we currently do not consider real-time detections. However, our approach may have potential real-time applications. Therefore, it is important to consider techniques that enable this extension possibility. Moreover, one-class SVMs do not offer an intuitive integration with neural networks.

4

Methodology

This chapter describes the methodology used in this thesis for the detection of behavioral adaptations in driving. The constituent sections of this chapter should provide an answer to the research question: *How can the detection process be implemented to ensure that detected events consistently correspond to instances of behavioral adaptations?* The chapter first provides a high-level overview of the detector in section 4.1, after which the individual components of the detector are described in section 4.2.

4.1. High-level overview

The proposed detection system has the main objective of identifying instances where the driver's behavior has adjusted in response to the assistance system. To achieve this goal, the detector is constructed with two main components, which are visualized in Figure 4.1. These two components are a neural network for driver identification, which indicates whether the current behavior corresponds to the expected behavior of the driver; and a novelty detection technique, which indicates whether the current behavior is considered normal or abnormal. In addition to the detector's two components, the dataset that contains driving behavior, though not part of the detector, is important for achieving effective detections.

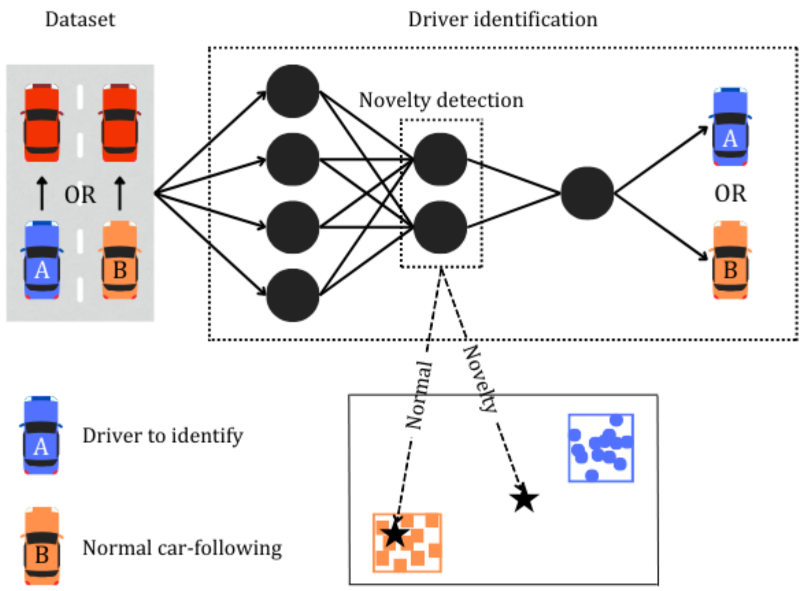


Figure 4.1: The high-level overview of the proposed detector. Features corresponding to the following vehicle are given as input to the neural network. The detector then has two different outputs. First, the network classifies whether this data corresponds to the expected behavior of the following vehicle. Second, the output of the final hidden layer before the output layer is extracted and, using the abstract-based novelty detection, the behavior is classified as normal or abnormal.

The neural network is designed to classify whether the input corresponds to the individual that needs to be identified. The idea is that when the behavior of the driver under identification has changed significantly from its expected driving behavior, the neural network will no longer classify its driving behavior correctly. This step allows us to indicate on an individual level whether any behavioral adaptations have occurred. An underlying assumption of this step is that the behavior of the driver remains consistent over time, with no significant drift, allowing any observed changes in behavior to be attributed to behavioral adaptations rather than natural fluctuations.

At the same time, novelty detection determines whether the detected change in behavior follows normal driving patterns or deviates from this behavior. By adding this extra classification step, we enrich the data provided by the detector. The behaviors and definitions used for behavioral adaptations and novelties are visualized in Figure 4.2.

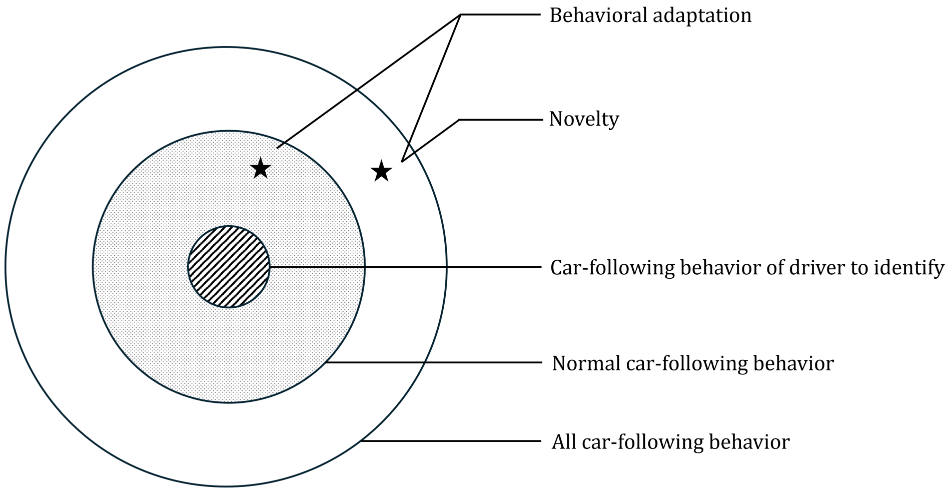


Figure 4.2: Schematic overview of the relations of the behaviors that can be encountered. Normal car-following behavior is defined as the range of driving behaviors that an average driver would exhibit, so without risky or unsafe behaviors. A novelty is defined as any input falling outside of normal car-following behaviors. Behavioral adaptation is defined as any input falling outside of behavior of the driver that we want to identify.

4.2. Building blocks

This section provides a detailed description of the dataset and the individual components that make up the detector. For each element, the design considerations are given and it is explained how the implementations of these designs should lead to an effective detection mechanism for behavioral adaptations. In subsection 4.2.1, subsection 4.2.2, and subsection 4.2.3, the dataset, the driver identification, and novelty detection are described, respectively.

4.2.1. Dataset

This section defines the considerations needed to create a fitting dataset for the detection of behavioral adaptations.

Novelty interpretation As mentioned in section 2.3, the novelty detection method is capable of detecting input from the neural network that has not been seen during training. This implies that the training dataset defines what inputs are detected as novelties during runtime. In this approach, we define the training dataset as the set of behaviors that define normal car-following behavior, where we assume that the expected behavior of the driver to identify is contained within this range of typical behavior. Therefore, if the novelty detection technique is trained on this dataset, any novelty will correspond to behavior outside of the distribution of normal car-following, thus indicating abnormal car-following behavior.

Stabilized behavior A design consideration is that *the dataset should include instances of stabilized driver behavior of the driver to identify while operating under the used assistance system.* As mentioned in section 3.1, when first introduced to an assistance system, the driver learns how to use the assistance system in various scenarios. After this learning phase, the behavior will stabilize before the driver adapts their behavior. So, stabilized behavior refers to periods in which the driver has gone through the learning phase of the assistance system and maintains consistent driving patterns. By including such instances in the dataset, the model can learn to recognize the typical behavior exhibited by the driver to identify when using the assistance system. This enables the detector to effectively detect any adaptations in driver behavior within this context.

Typical car-following Moreover, we need to be sure that typical behaviors in car-following situations are clearly defined, as defined by: *the modeling of normal behavior must be sufficiently comprehensive to ensure that any behavior falling within the bounds of typical behavior can be accurately identified as such*. This consideration ensures that the model can distinguish between typical and atypical behaviors with greater accuracy, thereby making sure that any detected novelty corresponds to atypical behavior, and any atypical behavior is consistently flagged as novelty. If the set of typical car-following behaviors is not sufficiently modeled, the detector might indicate novelties, while the behavior is actually contained in the range of typical car-following behaviors, thus indicating too many false positives.

4.2.2. Driver identification

The problem that the neural network tries to solve is a two-class classification problem with one class corresponding to a range of normal car-following behaviors and the other class corresponding to the driving behavior of a specific individual. The neural network provides the foundation for the detection system and its effectiveness in identifying the specific individual is vital for the accurate detection of both behavioral adaptations and their respective novelty. Therefore, the creation of the neural network requires careful design.

Input features

To achieve an effective identification of individuals, the network input should comprise features that accurately capture unique driving behaviors. These input features serve a dual purpose: to facilitate accurate driver identification and to provide valuable insights into behavioral adaptations. So, for the input of the network, we have the following design consideration: *the input features of the neural network should have the ability to accurately capture not only the unique driving style of an individual, but also to indicate any potential behavioral adaptations within driving behavior over time.* As described in section 3.3, driver identification is possible using features that can be collected from most modern vehicles. However, not all features used in these approaches can be modeled using the IDM (section 2.2), and as such cannot be used to model unique driving behavior in our simulations. The parameters that can be used to model car-following behavior are the desired velocity v_0, the maximum acceleration a, the comfortable deceleration b, the acceleration exponent δ, the minimum gap in standstill s_0, and the desired time-headway T. So, the features used in other approaches that cannot be modeled in our approach are the pedal pressure and operations, the steering angle, and lateral accelerations.

The set of features that we chose to represent unique driving behavior in car-following situations consists of speed, acceleration, jerk, and distance to the leading vehicle. Here, the jerk describes the rate of change in acceleration. The jerk was not mentioned in the other approaches, but we believe that it provides a more precise measure of sudden or rapid events in vehicle dynamics than acceleration alone. Therefore, the jerk is also added as a relevant input feature in this thesis. The speed and acceleration of the vehicle can be directly modeled using the IDM parameters for speed v_0, acceleration a, and braking b. The jerk is indirectly modeled, but changes as an effect of adapting a and b. The distance from the leading vehicle is also indirectly modeled using the parameter T for time headway. This set of features should also be able to capture behavioral adaptations, since the performance changes that we want to detect may manifest as increased accelerations, increased velocities, or heightened longitudinal deviations, all of which are behaviors identifiable through the designated set of features. Therefore, the selected features are in accordance with the defined design consideration.

Architecture

The architecture of the neural network should be designed such that the network can effectively extract the personal characteristics present in the driving data. We chose LSTM networks for our detector due to their simplicity and effectiveness in processing time series data, which aligns well with our driver behavior identification task. Although other methods have their advantages, we prioritized simplicity by using LSTMs. Many alternative approaches introduce additional complexities that can hinder interpretation and implementation of the detector. For the design of the neural network architecture, an important design consideration needed for efficient detection is: *the neural network should contain a layer that accurately describes the high-level information of the input*. According to the assumption made in the abstract-based novelty detection approach, layers at the end of the network capture this high-level information. We adapt a two-step sequential approach, where each layer extracts only the most relevant information from the previous layer's output. Consequently, this ensures that the final hidden layer captures the high-level information. First, we used an LSTM layer with 16 units to be able to extract sufficient information from the input. Second, the following LSTM layer further processes the information extracted by the first layer. This is implemented by giving this layer only 4 units, limiting the extraction capability, and forcing this layer to extract only the most informative features. Therefore, when the network is sufficiently trained, this final hidden layer should capture the high-level information from the input. An overview of this architecture is given in Figure 4.3

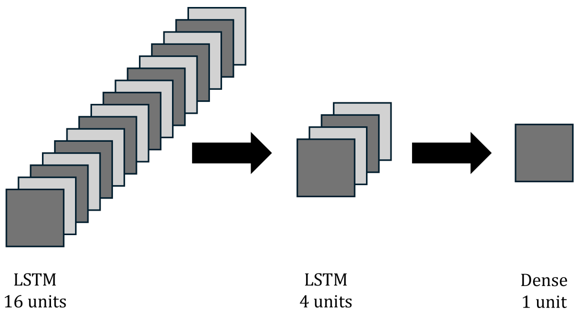


Figure 4.3: The architecture of the neural network used for the proposed detector, containing two LSTM layers and one fully-connected output layer.

Training procedure

With clear definitions of the design considerations for the input and architecture of the network in place, the neural network training procedure involves another element that decides the effectiveness of the detector. We can describe these aspects in the following design consideration: *the training procedure design must ensure that the network is properly trained for accurate driver identification and novelty detection*. There are three possible scenarios that we can have after training, underfitting of the data, overfitting of the data, and proper training, which have been visualized in Figure 4.4.

If the network is underfit on the training data, the spread of the training points will be high since the network has not learned to accurately group and separate the two classes. Consequently, behavioral adaptations can be misclassified as the driver to identify and vice versa. In novelty detection, the abstractions will not accurately capture the two classes, and a large part will not be correctly classified.

In the case of overfitting, although the training performance will be high, the testing of the known classes will yield low accuracy. However, since the network will only learn to classify training inputs as the driver to identify, any behavioral adaptation will not be classified as such, thus being accurately identified as behavioral adaptation. In the novelty detection, only exact training inputs will fall inside the boxes, therefore most points will be detected as novelties.

These cases indicate the need for an accurate training procedure. An element that already limits the overfitting ability of the network is the limited number of parameters defined in the architecture. Another straightforward technique is to keep a separate test set to independently track the evaluation throughout the training process, ensuring that the network performs well both in training and in testing. In the following chapters, these cases are used as a guideline to control and check the performance of the neural network.

4.2.3. Novelty detection

The second main component of the presented monitor is the novelty detection component. This stage is implemented using the abstract-based novelty detection approach. We can describe the goal and the design of novelty detection in the following design consideration: *the novelty detection technique*

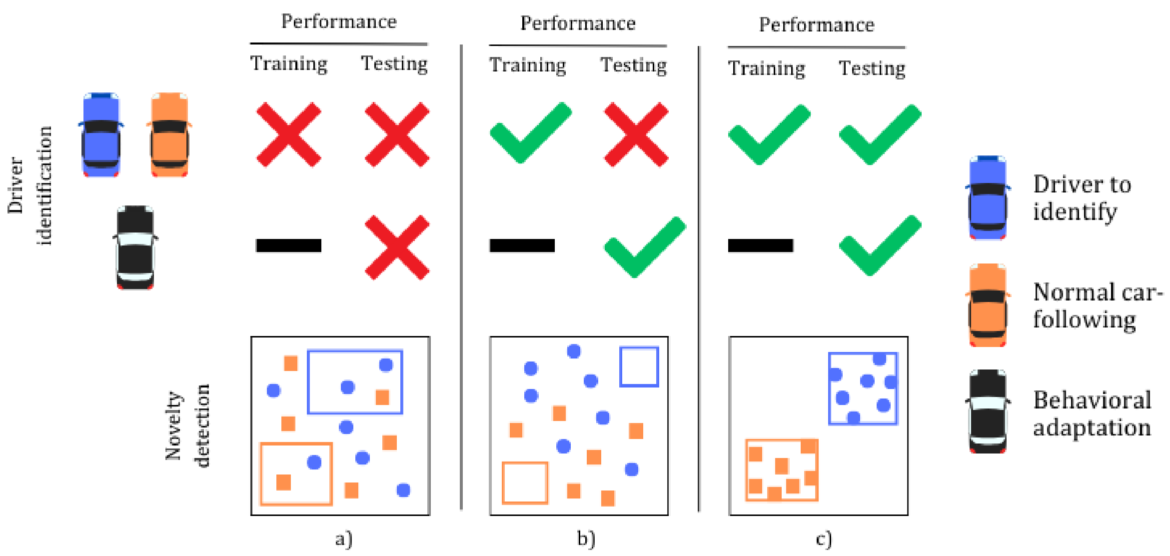


Figure 4.4: The three possible scenarios for training the detector: underfitting of the training data (a), overfitting of the training data (b), and proper training (c). The top row shows the performance of the neural network for each of the possible inputs during training and testing. The lower row shows the possible scenarios for the novelty detection monitor during testing.

requires that the formed clusters precisely encapsulate the key regions within the value space that distinctly define each respective class, ensuring a robust and accurate classification of novel instances. In the original paper, this is referred to as the convergence of abstractions [21], which assumes that the network is pretrained, and depends only on the amount of training data. They define convergence as the point where any new training input from known classes does not enlarge the abstraction anymore. So, to ensure that formed clusters accurately capture the key regions that define each class, we need sufficient training data to capture the entire range of behaviors that we could reasonably expect from drivers.

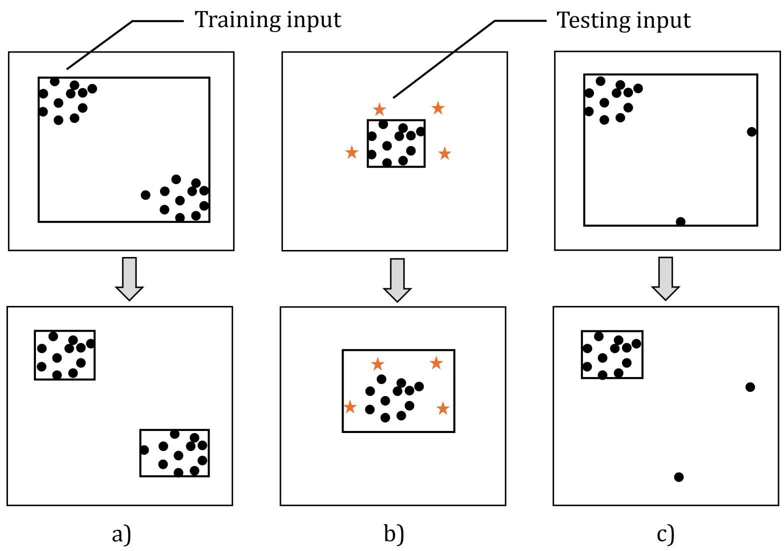


Figure 4.5: Techniques for fine-tuning the novelty detection, and the scenarios in which they are most useful. The improvements for cluster selection (a), expansion of abstractions (b), and outlier removal (c) are shown.

Fine-tuning parameters There are also additional techniques to control the creation of these clusters. These techniques are visualized in Figure 4.5. One technique is to set the number of clusters that are created. There is a possibility that the training points are not distributed in just one region but are distributed in separate regions of the value space. If we were to use just one cluster, all points in between these regions would also be considered to be part of the abstraction, whereas in reality they are not. Therefore, using a number of clusters representative for the number of regions leads to better capture of the key regions of the classes. We can also set the size of these clusters, where we can enlarge the abstractions, leading to more points falling within the boundaries during testing and evaluation. This leads to fewer false negatives, thus improving performance. Finally, we can control which points we take into account for the creation of the boxes, where it is possible to omit any outliers that cause the boxes to expand beyond representative regions of the distribution. In this thesis, the setting of the number of clusters, the expansion of the abstractions, and the omission of outliers is done empirically, based on the results of the detector. Examples of use cases for each of these techniques can be found in Appendix A.

Layer selection In the previous section, we outlined our choice for the layer to monitor, highlighting that the final layer before the output layer contains the high-level information of the input data. This layer, an LSTM layer consisting of 4 units, is thus used for both abstraction creation and novelty detection. To monitor an LSTM layer, we consider the final output of its units as the layer output. Consequently, since this monitored layer consists of 4 LSTM units, the resulting box abstraction will have 4 dimensions. This setup showcases the first approach to novelty detection of time series using the abstract-based approach to novelty detection.

5 Experimental setup

This chapter provides an overview of the experimental setup devised to address the research question: *To what extent do the events identified by the detector correspond to behavioral adaptations of the driver?* First, the collection of the dataset is described in section 5.1. Then, the experiments set up to answer the research question are given in section 5.2.

5.1. Dataset collection

This section provides the details for the collection of the dataset. In subsection 5.1.1 the simulation framework is described, in subsection 5.1.2 the setup of the simulation scenarios is given, in subsection 5.1.3 the modeled behaviors of the drivers in the scenarios are provided, and in subsection 5.1.4 the final preprocessing of the dataset is described.

5.1.1. Simulation framework

The dataset used in this study was collected using the CARLA (Car Learning to Act) driving simulator [52]. CARLA is an open source simulator designed for autonomous driving research. For the presented detector, to effectively detect behavioral adaptations, it is necessary to have both driver data before adaptations occur and during the shift to adapted behavior. Using the CARLA simulator, we have the possibility of simulating these behaviors and adaptations ourselves. This approach allows us to obtain a more tailored dataset and gives us greater control over the experimental conditions. Control over the environment is regulated using a framework [53], which provides readily available classes and implementations to precisely control car behavior, using an IDM and flexible scenario generation[1]. The adaptation of the framework we use in this thesis can be found on GitHub[2].

5.1.2. Scenarios

During the execution of the simulation, the scenario generation follows a cycle consisting of a series of stages. The simulation starts with the instantiation of vehicles, positioning the ego car behind the leading vehicle. Subsequently, the leading vehicle accelerates to the desired speed. Acting as the following vehicle, the ego car adheres to the specifications of the IDM, dynamically adjusting its speed and distance to ensure a safe trajectory behind the leading vehicle. This simulated behavior continues until a predetermined maximum number of time steps is reached. This maximum was set at 3000 to allow the vehicles to reach their desired speed and collect sufficient data about car-following behavior. Upon completion of each scenario, the vehicles are reset, and the IDM parameters are changed. The scenarios take place on the same map, as depicted in Figure 5.1. Per set of IDM parameters, the minimum simulation time set to capture the behavior of the vehicle was set to 10 minutes. Since the behavior of the following vehicle is stable under the same parameter set, this was believed to be sufficient. During the execution of the scenarios, the features of the following vehicle are collected every timestep T, where $T = 0.05$ seconds. The features that are collected are the speed v, acceleration a,

[1] https://github.com/MajidMoghadam2006/frenet-trajectory-planning-framework
[2] https://github.com/roudeelferink/thesis-simulate-driving

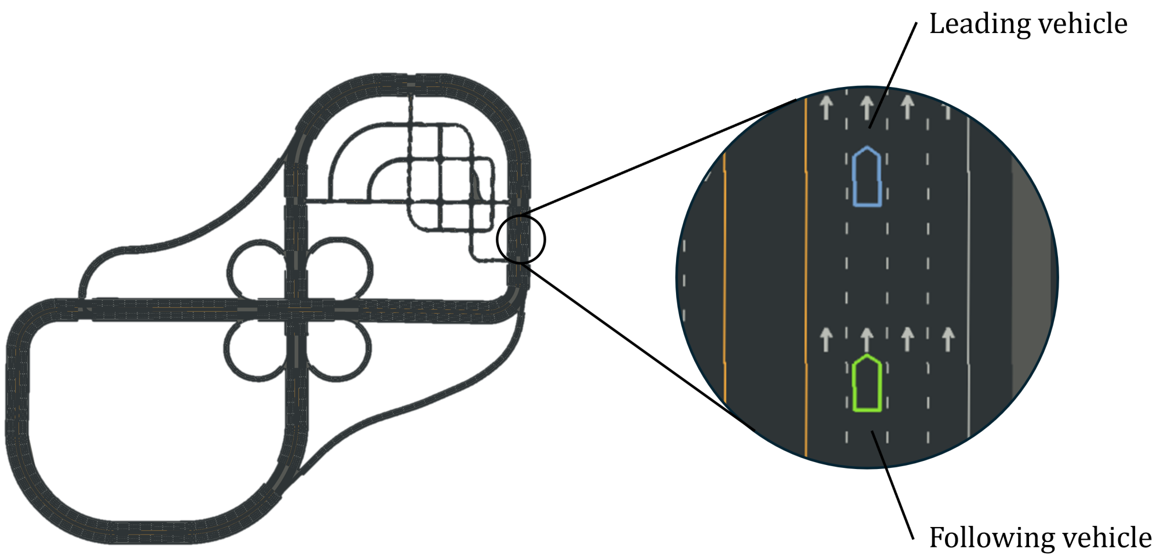


Figure 5.1: Road map and scenario setup used in the simulation.

jerk j, and distance to the leading vehicle d, as shown in Figure 5.2. The speed and distance to the leading vehicle are extracted directly from the simulator, while acceleration and jerk are calculated using Equation 5.1 and Equation 5.2. Here, t indicates the value in the current timestep and $t-1$ the value at the previous timestep.

$$a_t = \frac{v_t - v_{t-1}}{T} \quad (5.1)$$

$$j_t = \frac{a_t - a_{t-1}}{T} \quad (5.2)$$

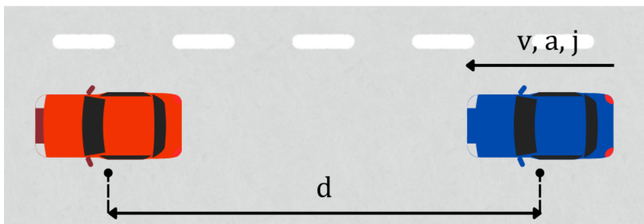


Figure 5.2: The features extracted from the simulator. Velocity v and distance d are extracted directly from the CARLA environment, acceleration a and jerk j are calculated using the given equations.

5.1.3. Behavior specification

Using the IDM, we can precisely model the behavior of the vehicle. We use the IDM to model the range of normal driving behavior of the following vehicle, the constant behavior of the leading vehicle, the following behavior of the driver to identify, and the behavioral adaptation of the driver to identify. An overview of the behaviors is provided in Figure 5.3. Each of these behaviors is described more specifically in the following paragraphs.

The range of normal car-following behaviors is set using ranges in the IDM parameters. We can extract realistic parameters that define normal behavior using data from real-life situations. This has been done in [54], where they used trajectory data to calibrate the IDM and predict the behavior such that it achieves the lowest possible error compared to the observed behavior. They used three data sets and reported the IDM parameters that achieved the lowest error. For the time-headway of the following

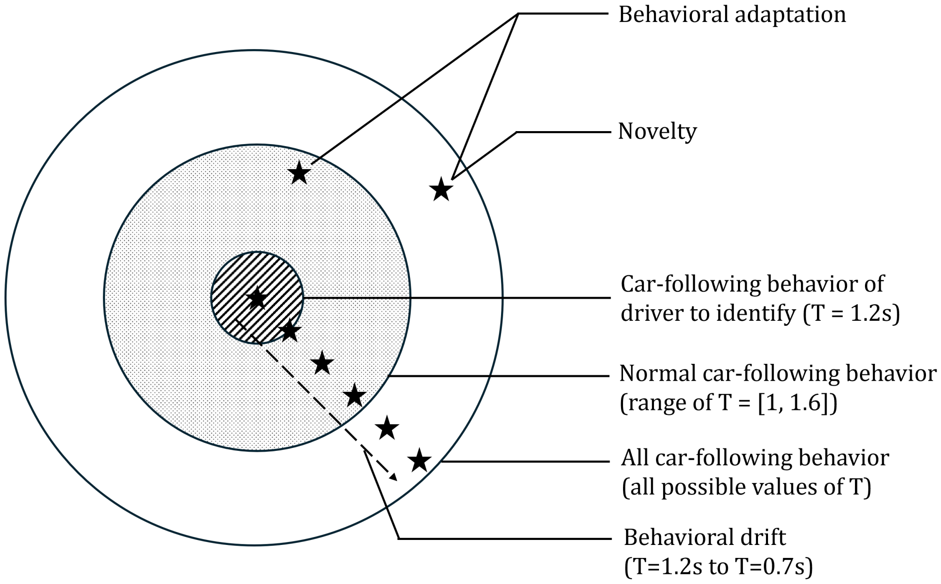


Figure 5.3: Schematic overview of the modeled behaviors. T stands for time headway, which corresponds to the time headway parameter of the IDM. The behavioral adaptation of the driver to identify is defined as a drift from their original behavior to a behavior outside of the range of normal behaviors. It shows that as the drift increases, the behavior should first be detected as behavioral adaptation, after which it will finally be detected as novelty as well.

vehicle, they found a range of 1 to 1.6 seconds, which is also used for this thesis. We model this in steps of 0.1 seconds, which means that we model 7 different behaviors for the time headway. However, for acceleration and deceleration, the values they reported caused unrealistically stable behaviors in our simulations, resulting in a lack of unique and identifiable behaviors. Therefore, a more extreme range of values was chosen for our experiments, defined by a range of 3 to 5 m/s^2 for acceleration and a range of 4 to 6 m/s^2 for deceleration. We did not simulate every possible combination of these parameters. Instead, we made pairs with the acceleration and deceleration to decrease the number of behaviors that we need to simulate. The three behaviors used to model the acceleration and deceleration range are given in Table 5.1. In combination with the range of time headway T, this leads to $7 * 3 = 21$ distinct behaviors that define typical driving behavior. The acceleration exponent was empirically set to 8, instead of the common value of 4, since the following vehicle could not keep up with the leading vehicle using a value of 4. Not all of these values are entirely realistic. However, the aim of the experiments is to demonstrate the detector's ability to identify behavioral adaptations and behaviors that fall outside of predefined normal ranges, which remains possible using these values. The final values for the range of the IDM parameters are given in Table 5.2.

Behavior	**Acceleration (m/s^2)**	**Deceleration (m/s^2)**
Behavior 1	3	4
Behavior 2	4	5
Behavior 3	5	6

Table 5.1: Table showing behaviors with corresponding acceleration and deceleration values.

The leading vehicle in the scenario has a distinctive behavior aimed solely at maintaining its target velocity, which dynamically shifts between values of 10.11 m/s (36.4 km/h), 11.11 m/s (40 km/h), or 12.11 m/s (43.6 km/h) at regular intervals. By periodically adjusting its target velocity within this range, the leading vehicle introduces variations in its behavior, contributing to a more realistic traffic environment. The other parameters are given in Table 5.2, but most of these values do not change the behavior of the leading vehicle in free-driving scenarios.

The driver to identify shows behavior contained within typical car-following behavior, maintaining a desired velocity of 13.89 m/s (50 km/h) to stay slightly ahead of the varying speed of the leading vehicle, ensuring that the leading vehicle does not drive away from the following vehicle. The other IDM parameters of the driver to identify are set with a minimum gap of 2.25 meters and a comfortable deceleration of 4 m/s^2. The maximum acceleration is set at 3 m/s^2 and desired time headway at 1.2 seconds. All of these values are contained within the defined range of typical driving behavior.

The behavioral adaptation emerging from the use of ADAS can also be modeled using the IDM. The chosen example adaptation examined in this thesis involves a reduction in the time headway of the driver to identify. This specific adaptation was chosen, because it is a behavioral adaptation that has been found to emerge in real life [55], and it is possible to model using the IDM, as shown in Figure 5.4. As behavioral adaptations do not occur instantaneously, modeling these changes should reflect their gradual nature. Thus, the behavior is adjusted incrementally, with a slight modification introduced every ten minutes of the simulation. Initially, the simulation uses the IDM parameters of the driver to identify. Subsequently, over ten steps, these parameters are changed to values that deviate from the driver's expected behavior, eventually drifting outside of defined normal car-following behavior. In this case, the time headway is decreased in increments of 0.05 seconds over ten steps, ultimately reaching a value of 0.7. This change of time headway, and the desired detections during this drift, are visualized in Figure 5.3.

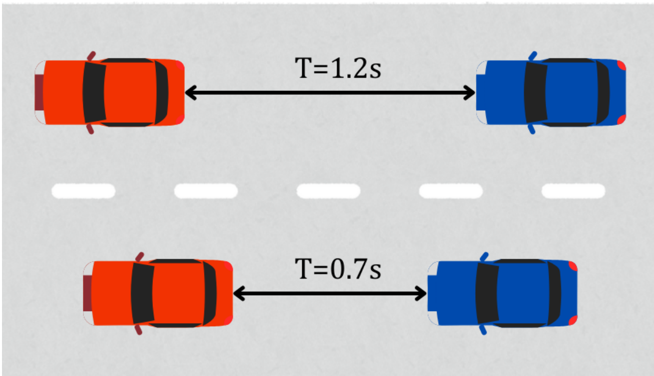


Figure 5.4: Decrease in in the time headway T of the IDM, and its effect on the driving behavior of the following vehicle.

Parameter	**Typical**	**Individual**	**Leading Vehicle**	**Behavioral Adaptation**
v_0	13.89 m/s (50 km/h)	13.89 m/s (50 km/h)	10.11-12.11 m/s	13.89 m/s (50 km/h)
s_0	2.25 meters	2.25 meters	2.25 meters	2.25 meters
b	4 m/s² to 6 m/s²	4 m/s²	2 m/s²	4 m/s²
a_{max}	3 m/s² to 5 m/s²	3 m/s²	1 m/s²	3 m/s²
T	1 s to 1.6 s (step: 0.1 s)	1.2 s	1 s	1.2 s to 0.7 s (step: 0.05 s)
δ	8	8	4	8

Table 5.2: IDM parameters for simulated behaviors.

Another aspect influencing the simulated behavior is the PID-controller. This controller ensures that the longitudinal error between the desired speed and the actual speed is as small as possible and that the lateral positioning matches the waypoints of the vehicle. These parameters were adapted from the original framework, where the values of $P = 40$, $I = 4$, $D = 0$ were used for the longitudinal controller and $P = 0.3$, $I = 0.0$, $D = 0.0$ for the lateral controller.

5.1.4. Preprocessing

After the data has been collected, the next step involves preprocessing the data to extract relevant segments for direct input into the neural network. To achieve this, we segment the time series data into fixed-length intervals of 100 data points, corresponding to approximately 5 seconds of driving. Segments where the vehicle is still accelerating towards the desired velocity are discarded, as are segments where crashes occur or where the following vehicle accelerates unrealistically fast. The processed data are then divided into training, testing, and runtime sets and the data features are normalized based on the data present in the training dataset. Both training and testing sets are used to train and test the neural network as well as the novelty detection monitor. The runtime set serves the purpose of evaluating how the monitor handles new inputs. In the labeled data, segments representing typical driving behavior are assigned the label 0, segments indicating the driver to identify are labeled 1, and segments demonstrating driver adaptation are labeled 2, where the latter group is exclusively present in the runtime dataset.

5.2. Experiments

This section presents the experiments that aim to answer our research question. We describe the setup for each experiment and outline the scenario in which the experiment can be considered a success. The first experiment to analyze behavioral adaptation detection is described in subsection 5.2.1. The second experiment to analyze novelty detection is described in subsection 5.2.2. The framework used for these experiments can be found on GitHub[3] and is an adaptation of the code[4] used for the experiments in [21].

5.2.1. Experiment 1: Detecting behavioral adaptations

The objective of this experiment is to assess the effectiveness of the developed detector in detecting the behavioral drift of the driver to identify, as the behavior drifts away from the driver's initial behavior, indicated by the most inner circle in Figure 5.3. So, in this experiment, we want the network to classify every behavioral adaptation as not corresponding to the driver to identify. The detection of behavioral adaptations is implemented using the neural network. Therefore, this experiment will focus solely on this component. The procedure of the experiment is as follows:

1. **Training Phase**: The neural network is trained on the training data, where the network learns to minimize the training loss.
2. **Testing Phase**: The trained neural network is tested on data containing unseen instances.
3. **Evaluation Phase**: The run data, containing instances of behavioral adaptations, is the final test of the network's effectiveness. The network's detections on the run data are analyzed to indicate whether the neural network can accurately identify instances of behavioral adaptations.

To get a more complete overview of the performance of the neural network, we repeat this procedure 10 times. This should indicate whether the network consistently learns to correctly classify the instances. We use the training and testing phase to show whether the network has been properly trained, as explained in Figure 4.2.2. The accuracy of the model is collected to assess the model's ability to distinguish between typical driving behavior and the individual's driving behavior. During the evaluation, we want to indicate the accuracy and sensitivity of the neural network. The accuracy indicates the reliability of the detection, as the behavior starts to drift. The sensitivity additionally indicates the point in the drift from which the behavioral adaptations start to be detected.

5.2.2. Experiment 2: Detecting unusual adaptations

The objective of this experiment is to evaluate the effectiveness of the novelty detection technique in detecting the behavioral drift of the driver, as behavior drifts outside the range of normal car-following behaviors. This is shown in Figure 5.3 as the drift of the points into the outer layer. This experiment follows the same three phases as before, now adapted for the novelty monitor:

1. **Training Phase**: The novelty detection monitor is trained on the training data. The training process involves fine-tuning model parameters to create boxes that best fit the training data, as explained in subsection 4.2.3.
2. **Testing Phase**: The trained monitor is tested using unseen instances. This should provide information about the generalizibility of the novelty detection.

[3]https://github.com/roudeelferink/thesis-monitoring-driving
[4]https://github.com/VeriXAI/Outside-the-Box/

3. **Evaluation Phase**: In the evaluation phase, the modeled behavior drift is given as input to the detector. The results should indicate whether the behavior of the driver is detected as unusual when it drifts outside of the predefined normal driving range.

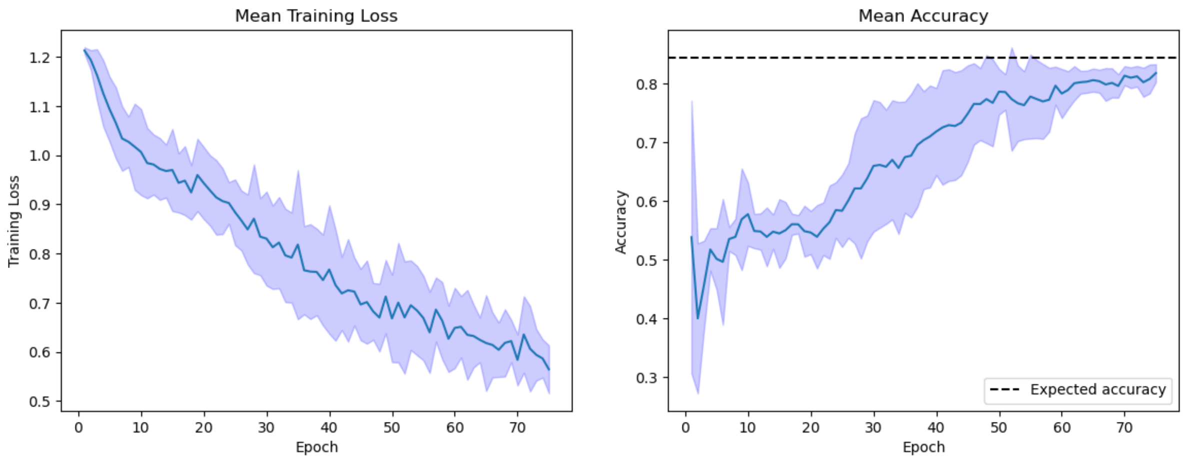
We collect the detector's outcomes and categorize them as true positives, false positives, true negatives, and false negatives. Again, we want to indicate the accuracy and sensitivity of the component. In this experiment, success is defined by the detector's ability to accurately detect unusual behaviors as novelties when the driving behavior drifts outside the range of normal driving behavior.

6 Results

This chapter presents the results of experiments that aim to evaluate the performance of the detector. In section 6.1, the results for the first experiment are presented, indicating the detector's performance in detecting behavioral adaptations. Then, in section 6.2, the results for the second experiment are presented, which indicate the detector's performance in detecting abnormal behaviors.

6.1. Results experiment 1: Detecting behavioral adaptations

This section presents the results of our experiment aimed at indicating the monitor's ability to detect behavioral adaptations. The initial training phase of the neural network is visualized in Figure 6.1. These figures show the training loss and accuracy of the network over 10 training runs. The parameters used in the training of the network can be found in Appendix A.



(a) Mean training loss of the neural network.

(b) Mean accuracy of the neural network.

Figure 6.1: Loss and accuracy during training of the network over 10 training procedure runs with the same parameters. The area around the line indicate the standard deviation over the 10 runs.

These figures show that, under the same parameters, the network consistently learns to accurately classify inputs. It is important to note here that the final accuracy achieved during training does not accurately capture the network's capability to make a distinction between the driver to identify and other behaviors. As visualized in Figure 4.2, the driver to identify is contained within the normal car-following behavior. This means that there is some overlap between the two classes, which makes it impossible to achieve perfect accuracy. This is further elaborated on in Appendix A. The figure shows that the network comes close to the expected accuracy, which is the accuracy we can reasonably expect the network to achieve with the overlapping behaviors.

In the testing phase, we put unseen data from the two training classes in the network. The results of this phase are given in Table 6.1. This table shows that the networks have the ability to generalize to unseen data points from the same class. Since both training and testing performance are high, this indicates that the network is properly trained. However, there are notable differences in accuracy. Especially in runs 2 and 5 we see that the network was not able to generalize to the new data.

Table 6.1: Final testing loss and accuracy over 10 runs. The last row indicates the mean and standard deviation over the 10 runs.

Run	Loss	Accuracy
1	0.385	0.835
2	0.557	0.679
3	0.298	0.862
4	0.367	0.826
5	0.511	0.752
6	0.363	0.835
7	0.317	0.844
8	0.404	0.807
9	0.294	0.862
10	0.318	0.844
Overall	0.382 ± 0.089	0.815 ± 0.057

In the evaluation phase, the handling of behavioral adaptations was analyzed, for which the results are shown in Table 6.2. In this phase, only instances containing behavioral adaptations are given as input to the network. Since the goal is to detect behavioral adaptations, we want the neural network to classify all instances of behavioral adaptation as class 0, indicating that the behavior does not correspond to the expected behavior of the driver. Since evaluation data only contain behavioral adaptations, only true positives and false negatives are taken into account. The table shows that when the behavior is still similar to expected behavior, that is, $T = 1.15$ and $T = 1.10$, the neural network does not consistently detect these instances as behavioral adaptations. However, all the classes afterwards are detected effectively, indicated by the (near) perfect accuracy. So, the results show that as the behavior is drifting away from the expected behavior, more of the instances of behavioral adaptation are detected as such. Regarding sensitivity, we see that at $T = 1.15$ the behavioral adaptations begin to be detected. However, the large standard deviation indicates that this is not the case for each of the runs. The same result continues in the next step of the drift at $T = 1.1$. This indicates that, although some of the inputs are correctly detected as behavioral adaptations, the detector is not sensitive enough to detect small changes in the behavior of the driver.

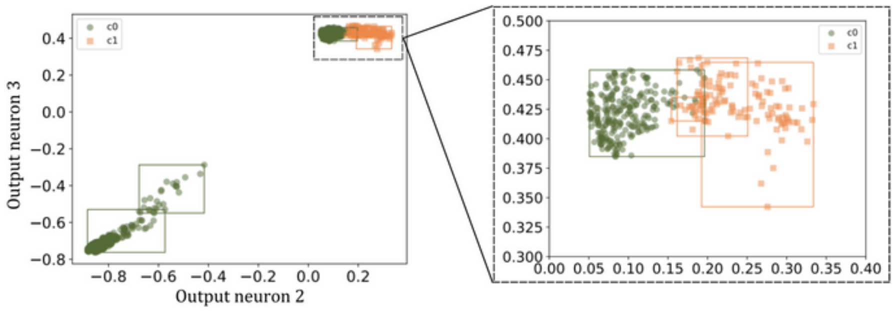
Table 6.2: Detection accuracy as the behavior drifts from T=1.15s to T=0.7s. A true positive occurs when the behavior is accurately identified as a behavioral adaptation. Conversely, a false negative indicates that the behavior, which should have been identified as a behavioral adaptation, is instead misclassified as typical behavior. Over 10 runs the mean and standard deviation of the detection is given for each step in the behavior drift.

Time headway (s)	True Positives	False Negatives	Accuracy (%)
1.15	20 ± 29	72 ± 29	21.30
1.10	62 ± 26	23 ± 26	72.94
1.05	85 ± 6	2 ± 6	97.70
1.00	94 ± 0	0 ± 0	100
0.95	91 ± 0	0 ± 0	100
0.90	98 ± 0	0 ± 0	100
0.85	102 ± 0	0 ± 0	100
0.80	101 ± 0	0 ± 0	100
0.75	90 ± 0	0 ± 0	100
0.70	97 ± 0	0 ± 0	100

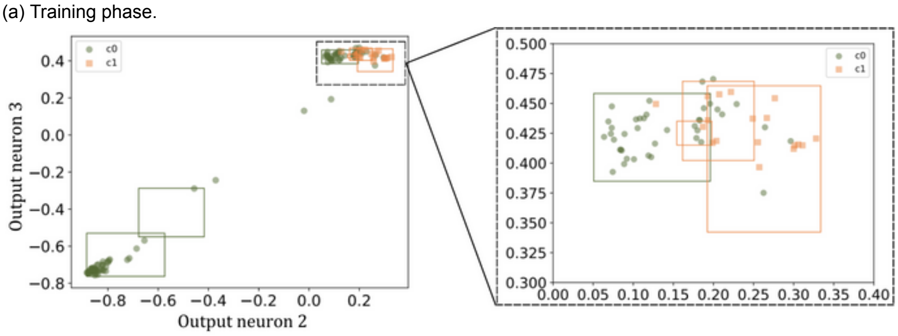
6.2. Results experiment 2: Detecting unusual adaptations

This section presents the results of our experiment aimed at indicating the detector's ability to detect unusual behavioral adaptations. We used training run 7 for the detection of novelties, since this run had a good performance and offered the most comprehensible visualizations. Other training runs can be found on GitHub[1].

The training and testing phase was used to set the parameters of the novelty monitor. Examples of 2-dimensional projections of the 4-dimensional abstraction are shown in Figure 6.2a and Figure 6.2b. These two dimensions were chosen since they best visualized the drift of the behavior. As can be seen in the figures, the distribution of class 0 is distributed over two regions. It was decided to use three clusters for the clustering of the abstractions. A third cluster was introduced to minimize the amount of empty space within the abstraction, which would have been larger if only one cluster had been used for the lower left region. Other techniques to fine-tune the abstractions were not necessary for this training run. Examples of other training runs and scenarios where fine-tuning could be useful are provided in Appendix A. The testing phase, shown in Figure 6.2b, indicates that almost all instances of class 1 are contained within the corresponding boxes. When it comes to class 0, we see that part of the instances is contained in the boxes of class 0 and part in the boxes of class 1. This can be explained by considering that class 0 has overlap with class 1, as visualized in Figure 5.3. Therefore, it is not surprising that some points are included in the abstraction of class 1. Most importantly, the fact that almost all test input falls within the trained abstractions indicates that the network was properly trained for this specific training run.



(a) Training phase.



(b) Testing phase.

Figure 6.2: The training (a) and testing (b) phase of the novelty monitor.

In the evaluation phase, the handling of behavioral adaptations was evaluated, for which the plot is shown in Figure 6.3. Interestingly, we see a clear drift of the distribution as the behavior adapts. It shows that at the end of the behavior drift, the instances clearly fall outside of the created abstractions. To get a better overview of the types of instances that fall outside the box, the behavioral instances are also shown in Figure 6.4, where the instances are grouped corresponding to their time headway. This figure more clearly shows the distribution of points throughout the drift. This allows us to indicate the point in the drift where the instances are detected as novelties. Ideally, what should happen is that from $T < 1$ onward the points should be detected as novelties, since that is the point where the behavior falls outside of the predefined normal range (see Figure 5.3). In fact, this is what we see in the figure,

[1]https://github.com/roudeelferink/thesis-monitoring-driving

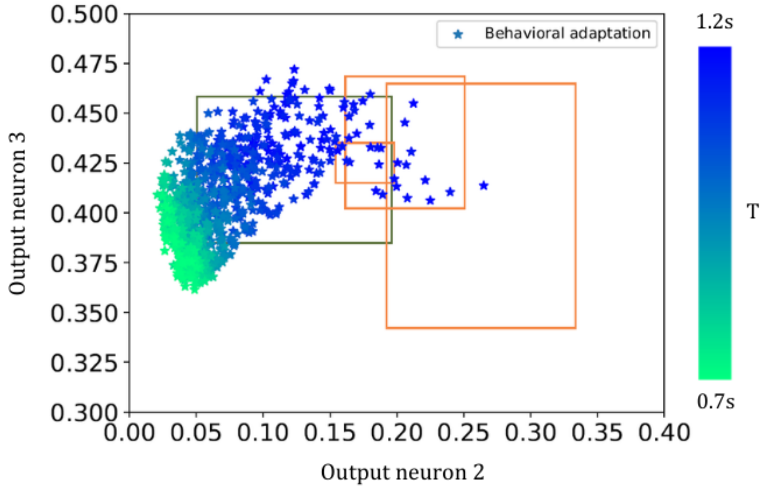


Figure 6.3: The behavior drift of the driver to identify from T=1.2s to T=0.7s, as captured by the novelty detection.

where the first instances drift outside of the abstraction at $T = 0.95$.

The qualitative analysis underscores the results found in this experiment. Table 6.3 presents the detection rate per class. The time headway $T = 1.15$ to $T = 1$ can still be considered normal behavior, since they fall within the defined boundaries of typical car-following behavior. All classes afterward are the classes that fall outside these boundaries, meaning that they should be detected as novelties. The table shows that when the instances should not be detected as novelties, i.e. the first four rows, the instances are mostly not detected as such.

Only at $T = 1.15$ we find that instances are sometimes detected incorrectly. In our process of modeling normal driving behavior, we increased the parameter T in steps of 0.1. This means that during the training process for the monitor, we collected data for specific values such as $T = 1.1$ and $T = 1.2$, but not for intermediate values such as $T = 1.15$. During neural network training, the inputs where $T = 1.2$ overlapped with the driver to identify were misclassified. Consequently, these misclassified points were not included in the creation of the abstractions. The outer edge of the abstraction box was thus set using points from $T = 1.1$, which explains why intermediate values such as $T = 1.15$ fall outside the defined box.

The table also shows that as the behavior drifts outside of the predefined range, the accuracy gradually increases. Already at $T = 0.95$ a substantial part of the instances is detected as novelty, showing the detector's sensitivity in detecting behavioral adaptations outside of normal behavior. This shows that on a high level we can indicate a drift, when we consider all points for a specific value of T. However, the notable false negative rate indicates that, on an individual instance level, it is difficult to indicate whether a point is an unusual behavioral adaptation. These results indicate that the detector can indeed be used to detect unusual adaptations of the driver. Moreover, it is important that these detection outcomes are not analyzed per specific instance but rather over a longer period of time to be able to indicate a drift outside of typical driving behavior.

T (s)	True Positives	False Positives	True Negatives	False Negatives	Accuracy
1.15*	0	28	64	0	0.6957
1.10*	0	0	85	0	1.0000
1.05*	0	0	87	0	1.0000
1.00*	0	2	92	0	0.9787
0.95	33	0	0	58	0.3626
0.90	73	0	0	25	0.7449
0.85	90	0	0	12	0.8824
0.80	97	0	0	4	0.9604
0.75	90	0	0	0	1.0000
0.70	97	0	0	0	1.0000

Table 6.3: Detection performance for the selected training run. Rows with a star (*) are the behaviors that are in the range of typical driving behavior. This means that these should not be detected as novelty. The other rows correspond to behaviors outside the range and which should be detected as novelty.

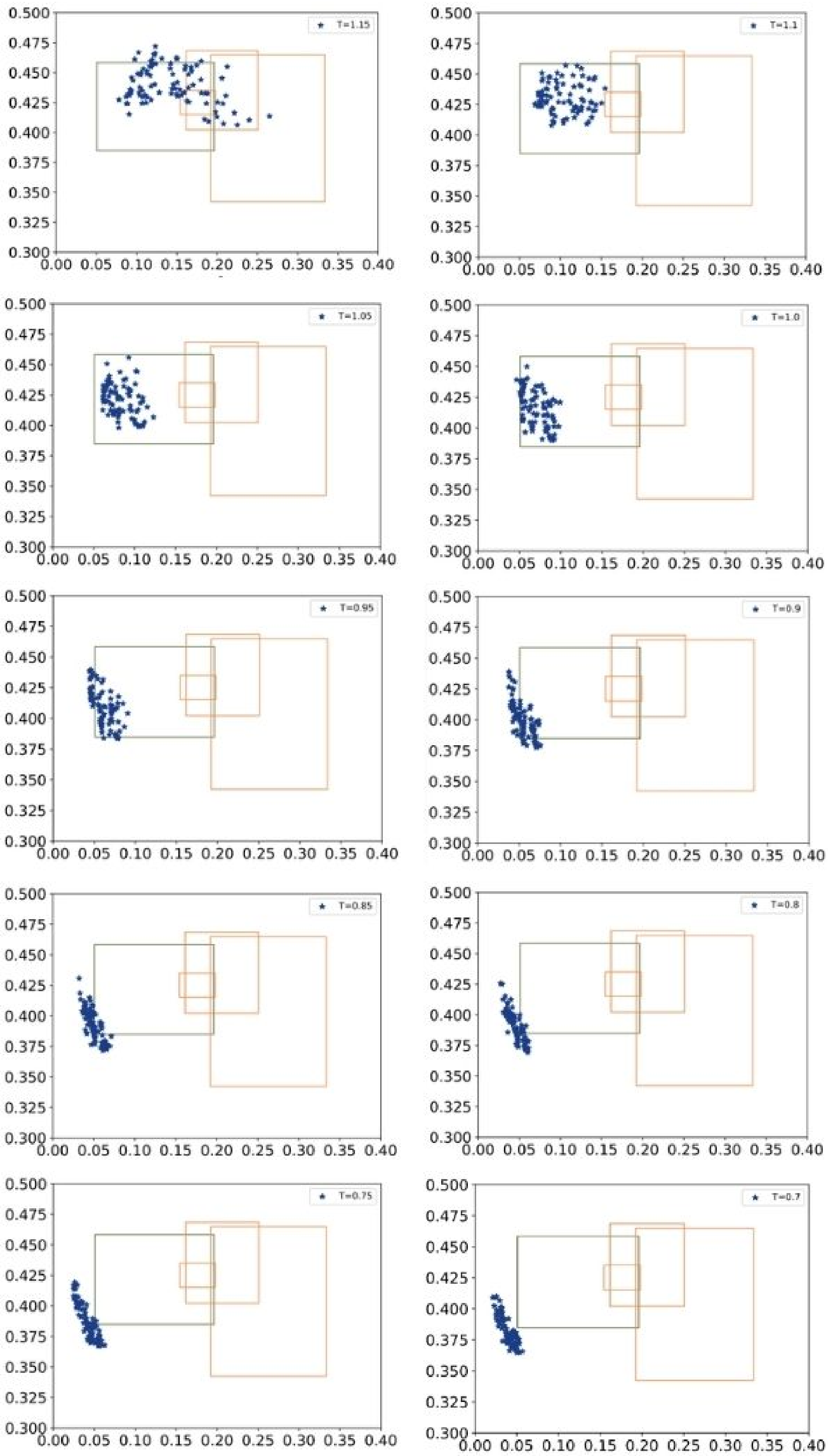


Figure 6.4: The distribution of the instances containing behavioral adaptations for each step from T=1.15s to T=0.7s.

7 Discussion

The research question used to guide this research was: *How can long-term behavioral adaptations, as an effect of driving with assistance systems, be effectively detected using the abstract-based approach to novelty detection?* The two sub-questions were formulated as:

1. How can the detection process be implemented to ensure that detected events consistently correspond to instances of behavioral adaptations?
2. To what extent do the events identified by the detector correspond to behavioral adaptations?

This thesis showed that the detection process can be implemented using a neural network for driver identification, combined with a novelty detection technique for behavior classification. Several design considerations were presented, indicating elements that need to be taken into account for effective detection. These included the creation of the dataset to ensure we capture meaningful novelties; the modeling of the neural network, to select relevant input features and create an effective architecture; and the setup of the novelty detection, which ensured that the abstractions capture only the key regions of the distribution.

The results presented in this thesis showed that the detector is effective in detecting simulated behavioral adaptation in a controlled setting, although the sensitivity of the detector could be improved. This indicated the potential to detect behavioral adaptations using a neural network. We also found that the abstract-based approach to novelty detection is an effective approach for the considered setting, where it could accurately capture meaningful novelties outside of predefined ranges of driving behavior. An important insight here was that only a high-level analysis of the detections will indicate this behavioral drift, while on a low level not every unusual behavior will be detected as such.

7.1. Assumptions & limitations

Although the results provide some interesting insights, we must consider the assumptions and scope set for this thesis. These assumptions give a good basis for obtaining meaningful results, but they also introduce constraints on the realism of the detector. Therefore, for each of the assumptions made, we briefly list how these constrain realism and how the detector could be adapted to eliminate these assumptions.

First, we only focus on car-following situations, since this is the most common driving behavior in everyday driving. This scope does not severely limit the applicability of the detector, since car-following is already a ubiquitous behavior in driving, and only focusing on this behavior should provide meaningful insights into behavioral adaptations. Therefore, it is suggested that any approaches that extend this detector first maintain this scope, before moving on to other behaviors on roads, such as overtaking, lane-changing, or signaling.

At this point in the research it is most valuable to eliminate the other assumptions made in this thesis. The most important assumptions that need attention are the exclusion of weather and road conditions, the constant speeds at which the experiments are executed, and the exclusion of noisy sensors. All of these factors complicate the training process of the neural network for driver identification, as more

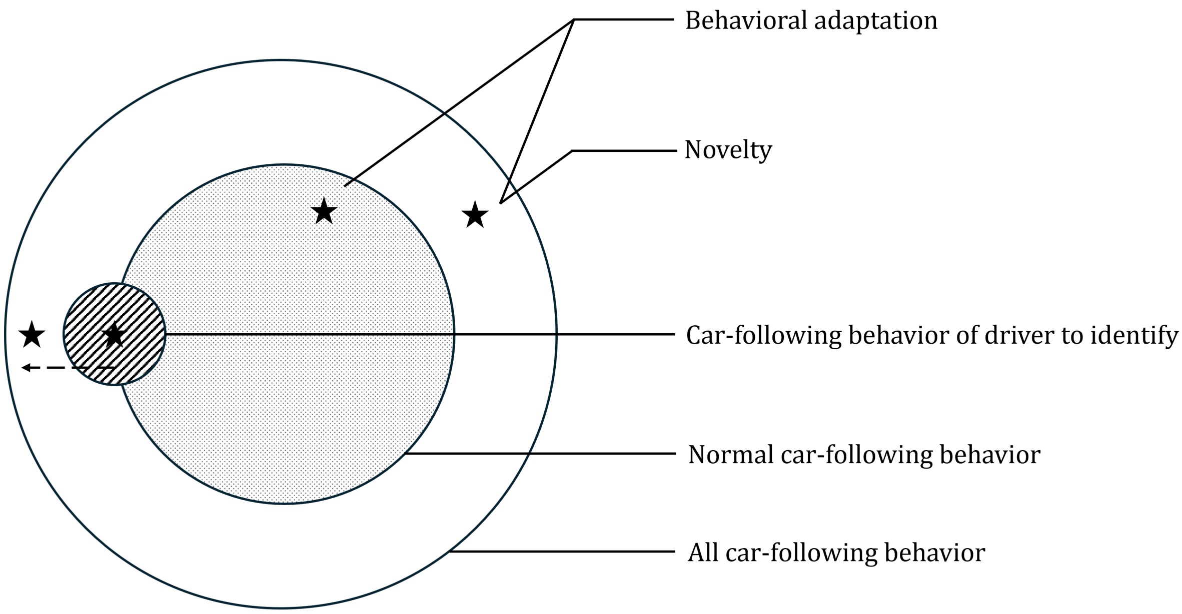


Figure 7.1: Scenario where initial behavior of driver to identify is not contained in range of normal car-following behavior. This could cause the behavior drift to still be classified as the driver to identify, since it is the closest class. In this case, the neural network would give a misclassification, but the novelty should still be detected.

factors must be taken into account. This also means that we need more data on driving under various environmental conditions to obtain an accurate and reliable identification. This data collection could present a significant challenge in adapting to real-life applications. The main challenge is to obtain a dataset that comprehensively captures all possible normal driving behaviors. This requires data for every possible normal behavior in every scenario, which is difficult to collect. Moreover, interpretations of ”normal” driving behavior vary widely among individuals, making it challenging to establish a general definition.

Another main assumption made in the design of the detector was that ”the behavior of the driver remains consistent over time, with no significant drift, allowing any observed changes in behavior to be attributed to behavioral adaptations rather than natural fluctuations”. However, in a realistic scenario, it might be possible that a driver adapts their behavior not because of the use of an assistance system but because of, for example, fatigue, distraction, or psychological factors. Finding a distinction between these types of adaptations and actual behavioral adaptations to the assistance system is not trivial, since they can have similar effects on driving performance. However, a promising insight here is that while natural fluctuations have a temporal nature, behavioral adaptations should have a more lasting effect on the behavior of drivers. Therefore, we could use this insight to create a threshold that indicates the point where we consider the detected changes in behavior to be a behavioral adaptation rather than a natural fluctuation.

The last assumption made in this thesis was that the behavior of the driver to identify is first contained within the range of normal behaviors. This assumption ensured that when the behavior of the driver to identify started to drift, it would be detected as normal driving behavior. Nonetheless, if the behavior of the driver to identify is not contained within the range of normal driving behavior at first, it might happen that any behavior drift of the driver might not get detected. This scenario is visualized and explained in Figure 7.1. This is a scenario that needs attention and for which a solution needs to be made that prevents the drift from being misclassified.

7.2. What can be detected?

The results have shown effectiveness in indicating a linear drift toward a shorter time-headway of the driver under consideration. But what other behavioral adaptations could our detector potentially detect, and what are the considerations needed to detect these behavioral changes? This section will try to provide some intuition for the detector’s capability by answering these questions.

The design consideration for the input features of the neural network stated: *the input features of the neural network should have the ability to accurately capture not only the unique driving style of an individual, but also to indicate any potential behavioral adaptations within driving behavior over time.* So, whenever we want to detect a certain behavioral adaptation, we need to make sure that the input feature that captures this behavioral adaptation is also informative for driver identification. If this input feature is not informative for driver identification, then during the training process the neural network will learn to ignore that input feature and map it to zero. Consequently, we will not be able to detect any changes in this input feature. So we can detect only those behavioral adaptations that can be detected with one of the input features, where this input feature should also be informative for driver identification.

Moreover, what happens if the behavior drift, unlike the modeled drift in this thesis, is not linear? We believe that it would still be accurately detected since we do not make any assumptions on the type of drift we will see. The detector is general enough to detect any type of behavior change of the driver to identify, and the direction of the change does not impact the detection capability.

Then, we consider the case where the behavior does not necessarily get negatively affected but does fall outside of defined normal ranges. In the case considered in this thesis that would correspond to a time-headway larger than 1.6 seconds. In that case, the detector would identify it as both behavioral adaptation and novelty, but would not indicate a negative adaptation. This example indicates that the detector does not necessarily only detect negative behavioral adaptations but also any change from normal driving behavior. This indicates that it is still important to manually analyze the detected events to indicate which type of behavioral adaptation has been detected.

An interesting insight from the results was that the distribution of the novelties seemed to follow a drift similar to the drift of the behavior. As the behavior drifted further away from normal behavior, the novelties also drifted further away from the abstraction. This insight is interesting because it seems that the final hidden layer did indeed capture some high-level information about the input. Therefore, there seems to be more information available in this layer than just an indication of a novelty. It could be interesting to investigate whether this also occurs in more complex settings. If this is the case, it would allow us to indicate not only a novelty but possibly also the nature and severity of the novelty, depending on where the novelty falls outside of the box. This is something that has great potential in this setting, but also in a more general setting where interpretations of novelties are vital, such as safety critical settings like healthcare, financial systems or cyber security [56].

Finally, the detector is designed to contribute to long-term research on behavioral adaptations, but can it also be used for short-term research? We believe that this would be difficult since, in order to have accurate detection, we need to have sufficient amounts of data. In short-term research, we do not have enough data available to learn the identifiable characteristics of the driver. Consequently, we would get a scenario similar to the first scenario described in Figure 4.2.2, where the model is underfit and not properly trained. Therefore, the presented approach only finds application in long-term settings, where we have access to sufficient amounts of data.

7.3. Application areas

Considering the current state of the detector, it would be premature to state that it could be effectively applied to detect behavioral adaptations in real-life. However, if all modeling assumptions have been taken care of, the detector will find application in a number of different areas. If we extrapolate the findings of this thesis, the creation of a perfect detector for behavioral adaptations would be able to improve the analysis process of behavioral adaptations in FOTs. Here, the detector could indicate the moments where the behavior deviates from expected behavior while considering realistic observation periods and contexts. Furthermore, novelty detection of the detector could provide insight into the direction and severity of the drift. If we extrapolate the findings even further, we could, for example, imagine a detection system that is able to give on-the-fly feedback to the driver about their driving behavior, indicating the danger of their behavioral changes and allowing them to revise their behavior. This would be helpful in promoting safe driving practices.

8 Conclusion

8.1. Summary

In recent years, the increasing integration of advanced driver assistance systems (ADAS) in vehicles has encouraged research into how these technologies influence driver behavior over time. Understanding these long-term behavioral adaptations is crucial for improving the safety and effectiveness of ADAS. This thesis has contributed to this ongoing research by exploring a way to automatically detect long-term behavioral adaptations. A promising technique was an abstract-based approach to novelty detection. Therefore, the central research question guiding this thesis was: *How can long-term behavioral adaptations, as an effect of driving with assistance systems, be effectively detected using the abstract-based approach to novelty detection?* An approach was presented, where a neural network detects behavioral adaptations and novelty detection additionally classifies the behavior as normal or abnormal. The findings demonstrated that this detector successfully identified simulated behavioral adaptations in a controlled environment, highlighting its potential application in real-world settings. Furthermore, this thesis confirms the efficacy of the abstract-based approach to novelty detection in the context of long-term behavioral studies to indicate an abnormal drift of behavior.

8.2. Future directions

To extend and improve the detector, the following approaches are suggested. In order to have more accurate data collection, we need further research on the start and end of the learning phase after first introduction to ADAS. In subsection 4.2.1, we mentioned that we need stabilized behavior to get useful data for driver identification. Currently, it is not known exactly when these learning phases end and when behavior stabilizes. Therefore, research into the timing of these phases is important for the performance of the detector.

Another direction for future research is to look at automatic fine-tuning of the novelty detection technique. Here, research could focus on automatic clustering, outlier removal, and abstraction extension, as mentioned in subsection 4.2.3. Being able to automatically adapt the novelty detection technique to any situation is needed to apply the presented approach in real-life. Research could also focus on making the modeling assumptions of the detector more realistic by including environmental conditions or considering noisy sensors. A promising dataset in which this could be tested is FollowNet [25], which contains realistic car-following data collected from several well-known driving datasets. A challenge here is to find a definition for normal behavior, where normality can be a subjective concept.

A

Training specifics

A.1. Data insights

This section provides more in-depth insights into the training data that have been collected for this thesis. The identifiable differences are shown between the input features of the modeled behaviors. In addition, this section indicates the maximum classification accuracy that can be achieved using this dataset.

A.1.1. Behavior visualization

As input to the neural network during training, we have two classes: behavior of the driver to identify, and behavior corresponding to normal driving behavior. For each of these classes, we have the input features: speed, acceleration, jerk, and distance to leading vehicle. After collecting data from the simulator, these features should be representative of the specific driving behavior we want to identify. Here, we show whether this is actually the case.

Figure A.1 to Figure A.4 provide information about each of the input features and indicate the differences in input features between the two classes. Class 0 is the range of normal behavior and class 1 is the driver to identify. From these figures, it should be apparent that the input features speed, acceleration and jerk are not sufficiently informative for separating the two classes. This was an unexpected result of the data collection phase, where it was expected that a change of acceleration and brake parameters of the IDM would exhibit more distinctive behavior. Only for the distance to the leading vehicle, in Figure A.4, there is a clear difference between the two classes. Consequently, this has implications for the accuracy that we can expect from training the network on classifying these two classes.

A.1.2. Maximum accuracy

There is some overlap between the two classes used for classification. This means that it is impossible for the neural network to achieve perfect accuracy. We can precisely indicate this overlap by considering the IDM parameters used to model each of the behaviors. Since primarily the distance to the leading vehicle appears to be informative for the classification, only the overlap is examined for this IDM parameter. We find that 15.5% of the normal driving range overlaps with the driver to identify. Therefore, the accuracy that we can reasonably expect the network to achieve during training is only 84.5%. This has no further consequences for the performance of the detector. However, it is important that each of these overlapping instances is classified as the driver to identify. If this is not the case, any overlapping behavior that corresponds to the driver to identify might get misclassified, thus incorrectly indicating a behavioral adaptation.

A.2. Novelty detection training

In the training process of the novelty detection there are several techniques that we can apply to improve performance, as described in subsection 4.2.3. This section shows examples of use cases for each of these techniques, collected from several training runs of the neural network.

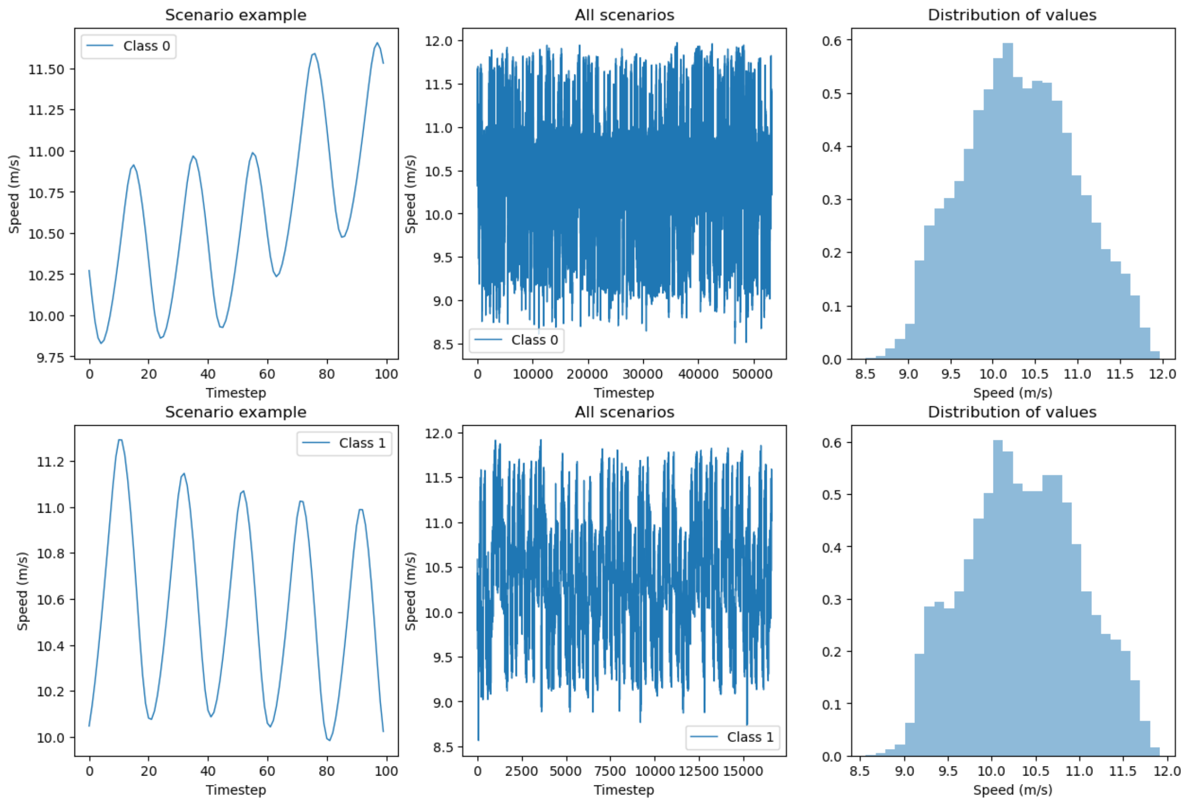


Figure A.1: Example of a scenario, all scenarios combined, and the distribution of values for the speed input feature.

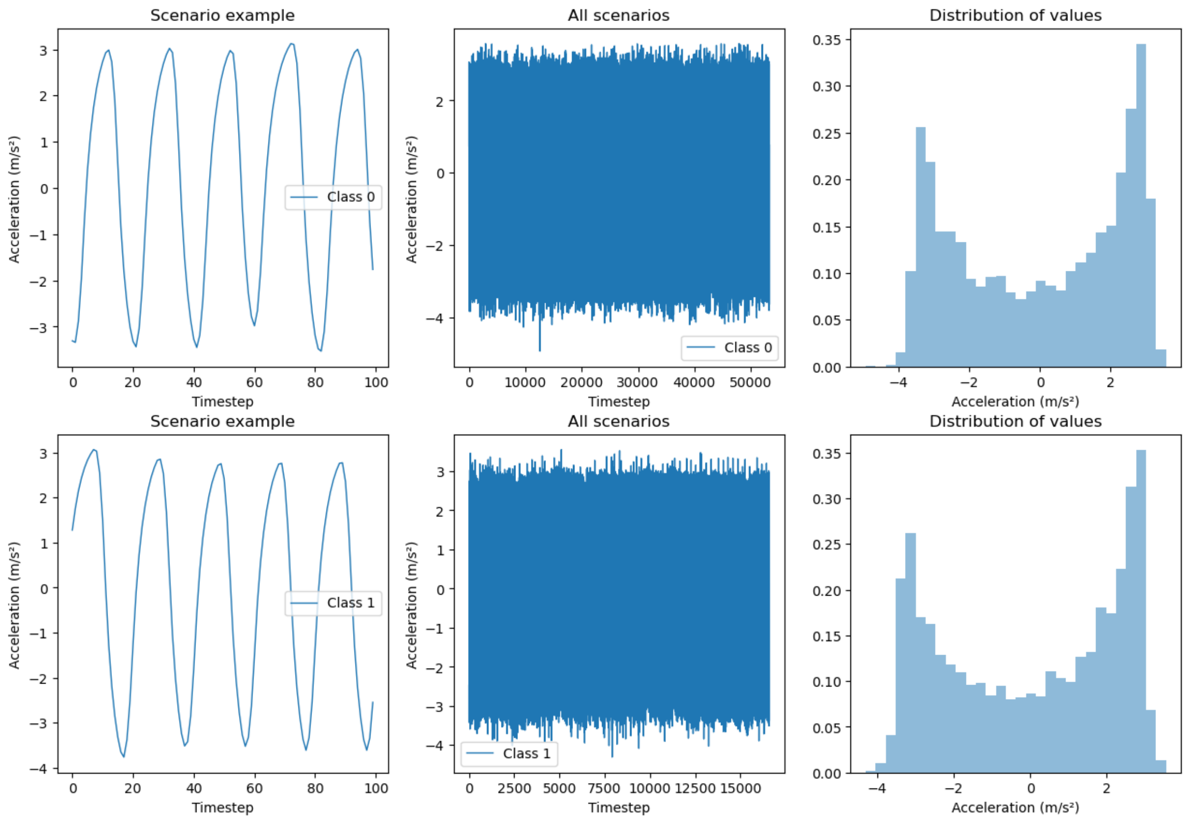


Figure A.2: Example of a scenario, all scenarios combined, and the distribution of values for the acceleration input feature.

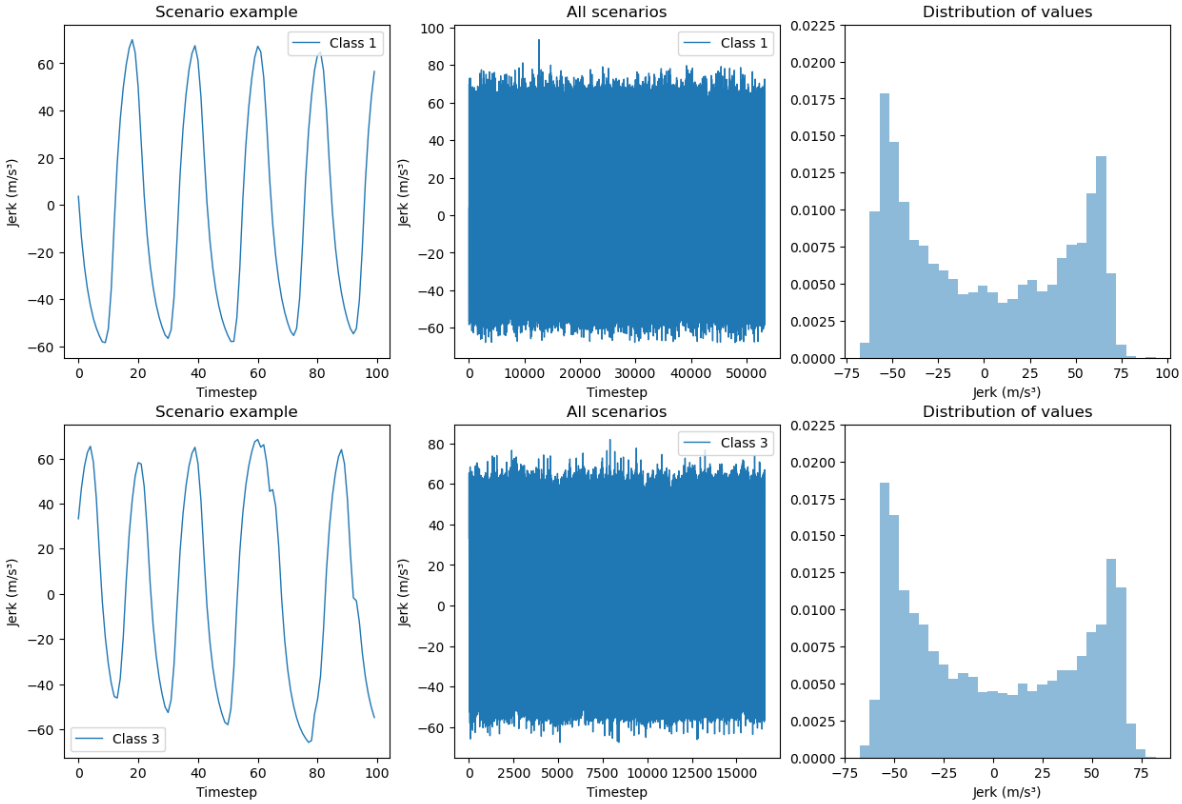


Figure A.3: Example of a scenario, all scenarios combined, and the distribution of values for the jerk input feature.

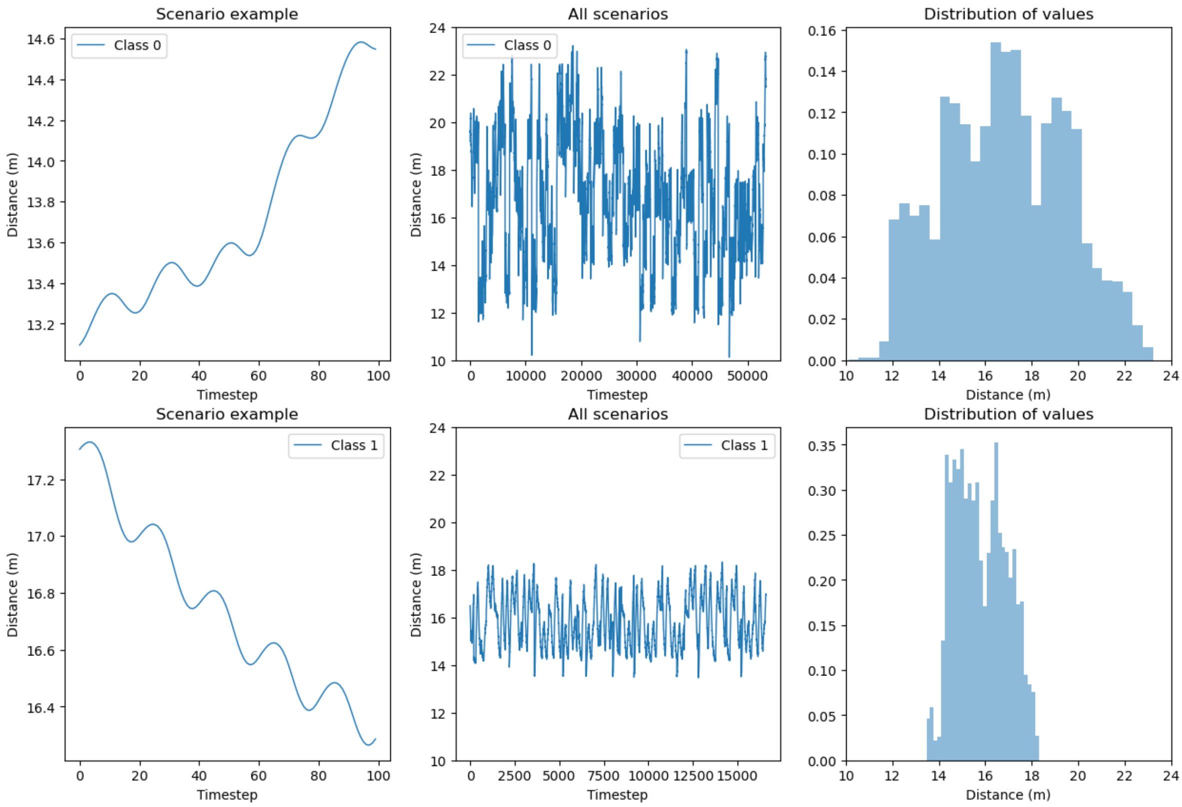


Figure A.4: Example of a scenario, all scenarios combined, and the distribution of values for the distance input feature.

Clustering Two use cases for clustering are shown in Figure A.5. These examples clearly show that the distribution of the training points is not contained in one region but rather in multiple distinct regions. This is especially visible for class 0. With fewer clusters, the clusters would capture more empty space, making the clusters less representative for the relevant regions of the distribution. With more clusters, we basically overfit on the training data, leading to a worse generalizability.

Box extension The goal of the box extension is to decrease the number of detected false negatives. A false negative occurs when normal behavior is classified as unusual behavior. Such a detection happens when an instance falls outside of the abstraction, while it should be contained. The use case for the box extension is visualized in Figure A.6. Here, we see that a test instance falls just outside of the abstraction. If the box had been extended to include this point, more of the points during the evaluation would have been correctly included in the abstraction. This shows how the box extension can be used to improve the performance of the detector.

Outlier removal Finally, outlier removal can be used in scenarios where outliers of the distribution lead to abstractions that capture unrepresentative regions of the distribution. Two examples of this scenario are shown in Figure A.7. The first example shows how a few outlier points expand the abstraction beyond the relevant distribution of the cluster. With the removal of these outliers in this cluster, the abstraction will capture the relevant region of the distribution with greater precision. The second example shows a cluster that contains only a few points. However, these points create a large box in an irrelevant part of the value space. Using outlier removal, these points can be removed to prevent them from being used in the creation of abstractions. A challenge of this fine-tuning process is that not every point far away from the main points of a distribution should be considered an outlier. An example of this can be found in the second example of Figure A.5. This example contains instances of class 0 that are still relevant, even though they are not near the two main clusters of the distribution.

A.3. Training parameters

Table A.1 gives an overview of all the relevant parameters used in the training process of the neural network. Most of these parameters were empirically chosen. The class weights are included to compensate for the class imbalance in the dataset. The values of the weights represent the ratio of class labels in the dataset.

Table A.1: Neural network training parameters.

Parameter	Value
Loss function	Binary cross entropy loss
Optimizer	RMSprop optimizer
Learning rate	Max learning rate 0.001
Batch size	64
Epochs	75
Class weights	1 for class 0, 6 for class 1

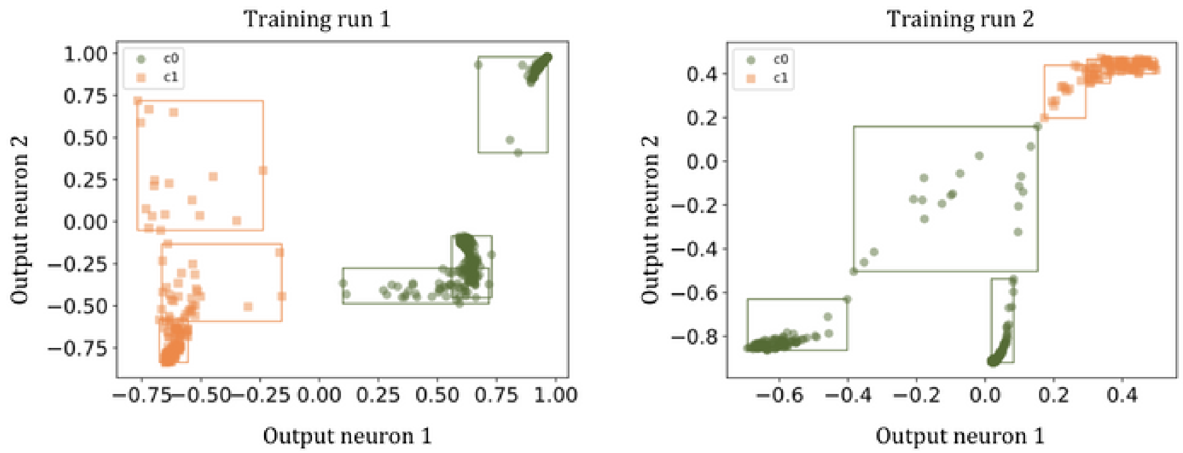


Figure A.5: Example use cases for cluster fine-tuning.

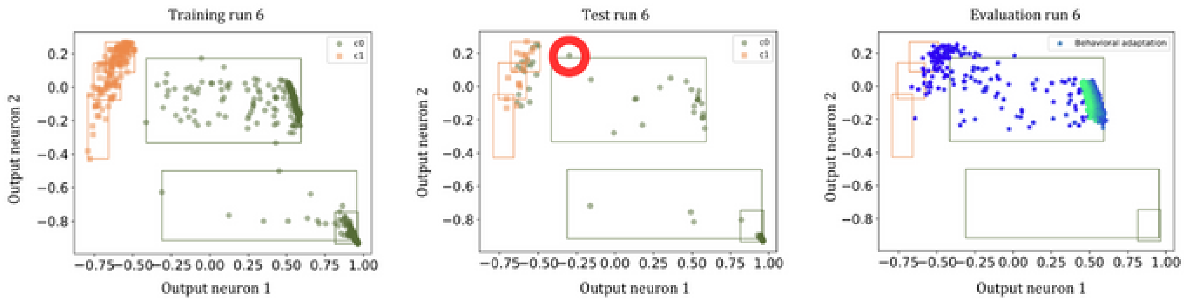


Figure A.6: Example use case for box extension.

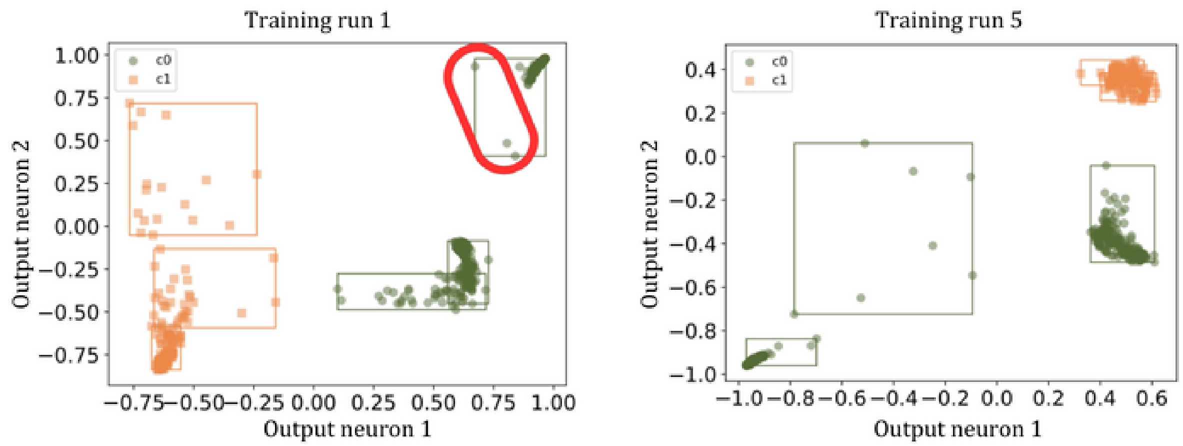


Figure A.7: Example use cases for outlier removal during training.

Bibliography

[1] *Automated driving revolution: Mercedes-Benz announces U.S. availability of DRIVE PILOT – the world's first certified SAE Level 3 system for the U.S. market*. en. Sept. 2023. URL: http://media.mbusa.com/releases/automated-driving-revolution-mercedes-benz-announces-us-availability-of-drive-pilot-the-worlds-first-certified-sae-level-3-system-for-the-us-market (visited on 05/15/2024).

[2] On-Road Automated Driving (ORAD) committee. *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*. en. DOI: 10.4271/J3016_202104. URL: https://www.sae.org/content/j3016_202104 (visited on 05/15/2024).

[3] Azim Eskandarian, ed. *Handbook of Intelligent Vehicles*. en. London: Springer London, 2012. ISBN: 978-0-85729-084-7 978-0-85729-085-4. DOI: 10.1007/978-0-85729-085-4. URL: http://link.springer.com/10.1007/978-0-85729-085-4 (visited on 12/04/2023).

[4] Aneesh Paul et al. "Advanced Driver Assistance Systems". In: Feb. 2016, pp. 2016–28–0223. DOI: 10.4271/2016-28-0223. URL: https://www.sae.org/content/2016-28-0223/ (visited on 06/04/2024).

[5] Alison Smiley. "Behavioral Adaptation, Safety, and Intelligent Transportation Systems". In: *Transportation Research Record* 1724.1724 (Jan. 2000), pp. 47–51. DOI: 10.3141/1724-07.

[6] M Hoedemaeker et al. "Behavioural adaptation to driving with an adaptive cruise control (ACC)". In: *Transportation Research Part F-traffic Psychology and Behaviour* 1.2 (Dec. 1998), pp. 95–106. DOI: 10.1016/s1369-8478(98)00008-4.

[7] Timo Melman et al. "Does haptic steering guidance instigate speeding? A driving simulator study into causes and remedies". In: *Accident Analysis & Prevention* 98 (Jan. 2017), pp. 372–387. DOI: 10.1016/j.aap.2016.10.016.

[8] Christina Rudin-Brown et al. "Behavioural Adaptation to Adaptive Cruise Control (ACC): Implications for Preventive Strategies". In: *Transportation Research Part F-traffic Psychology and Behaviour* 7.2 (Mar. 2004), pp. 59–76. DOI: 10.1016/j.trf.2004.02.001.

[9] Vishal C. Kummetha et al. "Analysis of the effects of adaptive cruise control on driver behavior and awareness using a driving simulator". In: *Journal of Transportation Safety & Security* 12.5 (May 2020), pp. 587–610. DOI: 10.1080/19439962.2018.1518359.

[10] Niek Beckers et al. "Drivers of partially automated vehicles are blamed for crashes that they cannot reasonably avoid". In: *Scientific Reports* 12.1 (Sept. 2022). DOI: 10.1038/s41598-022-19876-0.

[11] Farida Saad. "Some critical issues when studying behavioural adaptations to new driver support systems". In: *Cognition, Technology & Work* 8.3 (Sept. 2006), pp. 175–181. DOI: 10.1007/s10111-006-0035-y.

[12] Marieke H. Martens and Gunnar D. Jenssen. "Behavioral Adaptation and Acceptance". en. In: *Handbook of Intelligent Vehicles*. Ed. by Azim Eskandarian. London: Springer London, 2012, pp. 117–138. ISBN: 978-0-85729-084-7 978-0-85729-085-4. DOI: 10.1007/978-0-85729-085-4_6. URL: http://link.springer.com/10.1007/978-0-85729-085-4_6 (visited on 06/04/2024).

[13] Barbara Metz et al. "Repeated usage of a motorway automated driving function: Automation level and behavioural adaption". In: *Transportation Research Part F-traffic Psychology and Behaviour* 81 (Aug. 2021), pp. 82–100. DOI: 10.1016/j.trf.2021.05.017.

[14] Husam Muslim et al. "Long-Term Evaluation of Drivers' Behavioral Adaptation to an Adaptive Collision Avoidance System." In: *Hum. Factors* 63.7 (June 2020), pp. 1295–1315. DOI: 10.1177/0018720820926092.

[15] Frank Lai et al. "The long-term effect of intelligent speed adaptation on driver behaviour". In: *Applied Ergonomics* 41.2 (Mar. 2010), pp. 179–186. DOI: 10.1016/j.apergo.2009.03.003.

[16] Erika E. Miller et al. "Behavioral Adaptations to Lane Keeping Systems: Effects of Exposure and Withdrawal:" in: *Human Factors* 61.1 (Feb. 2019), pp. 152–164. DOI: 10.1177/0018720818800538.

[17] Francesco Viti et al. "Driving behavior interaction with ACC: results from a Field Operational Test in the Netherlands". In: *IEEE Intelligent Vehicles Symposium* (June 2008), pp. 745–750. DOI: 10.1109/ivs.2008.4621199.

[18] Carey Borkoski et al. "Understanding the Impact of Technology_Do Advanced Driver Assistance and Semi-Automated Vehicle Systems Lead to Improper Driving Behavior". In: (Nov. 2019).

[19] Oliver Carsten et al. "Safety Assessment of Driver Assistance Systems". In: *European Journal of Transport and Infrastructure Research* (Nov. 2001). DOI: 10.18757/ejtir.2001.1.3.3666.

[20] *The FESTA Handbook*. fr. Mar. 2019. URL: https://www.connectedautomateddriving.eu/methodology/festa/ (visited on 05/28/2024).

[21] Thomas A. Henzinger et al. "Outside the Box: Abstraction-Based Monitoring of Neural Networks". In: Frontiers in Artificial Intelligence and Applications (Nov. 2019), pp. 2433–2440. DOI: 10.3233/faia200375.

[22] Konstantin Kueffner et al. "Into the unknown: active monitoring of neural networks (extended version)". en. In: *International Journal on Software Tools for Technology Transfer* 25.4 (Aug. 2023), pp. 575–592. ISSN: 1433-2779, 1433-2787. DOI: 10.1007/s10009-023-00711-4. URL: https://link.springer.com/10.1007/s10009-023-00711-4 (visited on 06/04/2024).

[23] Yvonne Barnard et al. "Field operational tests: challenges and methods". In: Proceedings of European Conference on Human Centred Design for Intelligent Transport Systems (Apr. 2010), pp. 323–332.

[24] Yvonne Barnard et al. "Methodology for field operational tests of automated vehicles". In: *Transportation research procedia* 14 (Jan. 2016), pp. 2188–2196. DOI: 10.1016/j.trpro.2016.05.234.

[25] Xianda Chen et al. "FollowNet: A Comprehensive Benchmark for Car-Following Behavior Modeling". In: *Scientific Data* (2023). DOI: 10.1038/s41597-023-02718-7.

[26] Yang Zhou et al. "Modeling Car-Following Behaviors and Driving Styles with Generative Adversarial Imitation Learning". In: *Sensors* 20.18 (2020), p. 5034. DOI: 10.3390/s20185034.

[27] Organisation for Economic Co-operation and Development. *Behavioural adaptations to changes in the road transport system: report*. eng. Road transport research. Paris: OECD, 1990. ISBN: 978-92-64-13389-1.

[28] Gunnar Jenssen and Gunnar Deinboll Jenssen. "Behavioural Adaptation to Advanced Driver Assistance Systems : Steps to Explore Safety Implications". In: (Jan. 2010).

[29] Martin Treiber et al. "Congested traffic states in empirical observations and microscopic simulations". In: *Physical Review E* 62.2 (Aug. 2000), pp. 1805–1824. DOI: 10.1103/physreve.62.1805.

[30] Mohammad Saifuzzaman and Zuduo Zheng. "Incorporating human-factors in car-following models: A review of recent developments and research needs". In: *Transportation Research Part C-emerging Technologies* 48 (Nov. 2014), pp. 379–403. DOI: 10.1016/j.trc.2014.09.008.

[31] Tom Alkim et al. "Field Operational Test "The Assisted Driver"". In: *IEEE Intelligent Vehicles Symposium* (June 2007), pp. 1198–1203. DOI: 10.1109/ivs.2007.4290281.

[32] Nengchao Lyu et al. "A field operational test in China: Exploring the effect of an advanced driver assistance system on driving performance and braking behavior". In: *Transportation Research Part F-traffic Psychology and Behaviour* 65 (Aug. 2019), pp. 730–747. DOI: 10.1016/j.trf.2018.01.003.

[33] Mingming Zhang et al. "SafeDrive: Online Driving Anomaly Detection From Large-Scale Vehicle Data". In: *IEEE Transactions on Industrial Informatics* 13.4 (Feb. 2017), pp. 2087–2096. DOI: 10.1109/tii.2017.2674661.

[34] Zhongyang Chen et al. "D 3 : Abnormal driving behaviors detection and identification using smartphone sensors". In: *Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks* (June 2015), pp. 524–532. DOI: 10.1109/sahcn.2015.7338354.

[35] Fabio Tango et al. "Real-Time Detection System of Driver Distraction Using Machine Learning". In: *IEEE Transactions on Intelligent Transportation Systems* 14.2 (June 2013), pp. 894–905. DOI: 10.1109/tits.2013.2247760.

[36] Jiadi Yu et al. "Fine-Grained Abnormal Driving Behaviors Detection and Identification with Smartphones". In: *IEEE Transactions on Mobile Computing* 16.8 (Aug. 2017), pp. 2198–2212. DOI: 10.1109/tmc.2016.2618873.

[37] Dajun Wang et al. "Risky Driver Recognition Based on Vehicle Speed Time Series". In: *IEEE Transactions on Human-Machine Systems* 48.1 (Feb. 2018), pp. 63–71. DOI: 10.1109/thms.2017.2776605.

[38] Darko Mitrović, Dejan Mitrovic, and D. Mitrovic. "Reliable method for driving events recognition". In: *IEEE Transactions on Intelligent Transportation Systems* 6.2 (June 2005), pp. 198–205. DOI: 10.1109/tits.2005.848367.

[39] Thitaree Tanprasert et al. "Combining Unsupervised Anomaly Detection and Neural Networks for Driver Identification". In: *Journal of Advanced Transportation* 2017 (Oct. 2017), pp. 1–13. DOI: 10.1155/2017/6057830.

[40] Matthias Matousek et al. "Detecting Anomalous Driving Behavior using Neural Networks". In: *2019 IEEE Intelligent Vehicles Symposium (IV)* (June 2019). DOI: 10.1109/ivs.2019.8814246.

[41] Nuttun Virojboonkiate et al. "Driver identification using histogram and neural network from acceleration data". In: *International Conference on Speech Technology and Human-Computer Dialogue* (Oct. 2017). DOI: 10.1109/icct.2017.8359893.

[42] Jingbo Yang et al. "Driver2vec: Driver Identification from Automotive Data". In: *arXiv.org* (2021).

[43] Toshihiro Wakita et al. "Driver identification using driving behavior signals". In: *Proceedings. 2005 IEEE Intelligent Transportation Systems, 2005.* (Oct. 2005), pp. 396–401. DOI: 10.1109/itsc.2005.1520171.

[44] Chiyomi Miyajima et al. "Driver Modeling Based on Driving Behavior and Its Evaluation in Driver". In: *Proceedings of the IEEE* (Jan. 2007). DOI: 10.1109/jproc.2006.888405.

[45] Sasan Jafarnejad et al. "Towards a real-time driver identification mechanism based on driving sensing data". In: *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)* (Oct. 2017), pp. 1–7. DOI: 10.1109/itsc.2017.8317716.

[46] Tung Kieu et al. "Outlier Detection for Time Series with Recurrent Autoencoder Ensembles". In: *International Joint Conference on Artificial Intelligence* (Aug. 2019), pp. 2725–2732. DOI: 10.24963/ijcai.2019/378.

[47] Sakti Saurav et al. "Online anomaly detection with concept drift adaptation using recurrent neural networks". In: *COMAD/CODS* (Jan. 2018), pp. 78–87. DOI: 10.1145/3152494.3152501.

[48] Dit–Yan Yeung et al. "Host-based intrusion detection using dynamic and static behavioral models". In: *Pattern Recognition* 36.1 (Jan. 2003), pp. 229–243. DOI: 10.1016/s0031-3203(02)00026-2.

[49] Padhraic Smyth and Padhraic Smyth. "Markov monitoring with unknown states". In: *IEEE Journal on Selected Areas in Communications* 12.9 (Dec. 1994), pp. 1600–1612. DOI: 10.1109/49.339929.

[50] Larry M. Manevitz et al. "One-class svms for document classification". In: *Journal of Machine Learning Research* 2.2 (Mar. 2002), pp. 139–154.

[51] Bernhard Schölkopf et al. “Support Vector Method for Novelty Detection”. In: *Neural Information Processing Systems* 12 (Nov. 1999), pp. 582–588.

[52] Alexey Dosovitskiy et al. “CARLA: An Open Urban Driving Simulator”. In: *arXiv: Learning* (Nov. 2017).

[53] Majid Moghadam and Gabriel Hugh Elkaim. “An Autonomous Driving Framework for Long-term Decision-making and Short-term Trajectory Planning on Frenet Space”. In: *arXiv: Robotics* (2020). DOI: 10.1109/case49439.2021.9551559.

[54] Arne Kesting et al. “Calibrating Car-Following Models by Using Trajectory Data: Methodological Study”. In: *Transportation Research Record* 2088.2088 (Jan. 2008), pp. 148–156. DOI: 10.3141/2088-16.

[55] Eva-Maria Eick and Guenter Debus. “Adaptation Effects in an Automated Car-Following Scenario”. en. In: *Traffic and Transport Psychology*. Elsevier, 2005, pp. 243–255. ISBN: 978-0-08-044379-9. DOI: 10.1016/B978-008044379-9/50175-8. URL: https://linkinghub.elsevier.com/retrieve/pii/B9780080443799501758 (visited on 06/04/2024).

[56] Marco A. F. Pimentel et al. “A review of novelty detection”. In: *Signal Processing* 99.99 (June 2014), pp. 215–249. DOI: 10.1016/j.sigpro.2013.12.026.