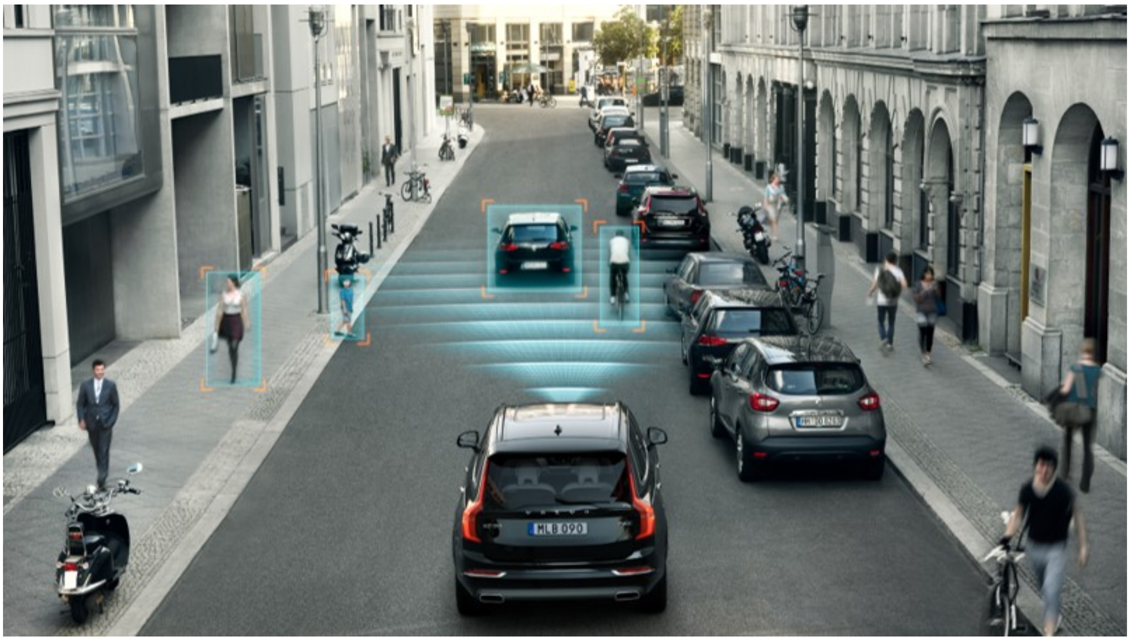
CHALMERS
UNIVERSITY OF TECHNOLOGY



Adaptive Driver Modelling for Forward Collision Warning Systems

An investigation of the correlation between behaviour at routine and near-collision driving situations

Master's thesis in Complex Adaptive Systems

OSCAR FORSMAN
JOHANNA WARNQVIST

DEPARTMENT OF MATHEMATICAL SCIENCES

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2021
www.chalmers.se

MASTER'S THESIS 2021

Adaptive Driver Modelling for Forward Collision Warning Systems

An investigation of the correlation between behaviour at routine and near-collision driving situations

OSCAR FORSMAN
JOHANNA WARNQVIST

Department of Mathematical Sciences
Division of Applied Mathematics and Mathematical Statistics
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2021

Adaptive Driver Modelling for Forward Collision Warning Systems
An investigation of the correlation between behaviour at routine and near-collision driving situations
Oscar Forsman & Johanna Warnqvist

Supervisor: Andreas Runhäll & Daniel Irekvist, Volvo Cars
Examiner: Torbjörn Lundh, Mathematical Sciences

Master's Thesis 2021
Department of Mathematical Sciences
Division of Applied Mathematics and Mathematical Sciences
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Visualisation of the City Safety system. Source: Volvo Cars, internal resources.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2021

Adaptive Driver Modelling for Forward Collision Warning Systems
An investigation of the correlation between behaviour at routine and near-collision driving situations
Oscar Forsman & Johanna Warnqvist
Department of Mathematical Sciences
Chalmers University of Technology

Abstract

In this work, driving behaviour is analysed with the purpose of finding connections between a drivers routine driving and their behaviour in collision and near-collision situations. The ambition is to improve the Forward Collision Warning (FCW) system on Volvo cars by taking information from previous driving situations of the current driver into account when determining the best timing for issuing a collision warning. The analysis is performed by means of feature extraction on multivariate time series data, containing measurements from various sensors. Using principal component analysis (PCA) and clustering methods such as k-means and DBSCAN, no connections relevant to the formulated aim could be found in the investigation. The conclusion drawn is that a more thorough evaluation of the available data is required. Removing parts of drive sequences that are not of interest or categorise the sequences into different scenarios can make the information more comparable and hence yield a better result. A more careful data cleaning of the available time series could also lead to an improvement.

Keywords: Forward Collision Warning, FCW, data analysis, clustering, tsfresh, PCA.

Acknowledgements

First and foremost, we would like to express our gratitude to our supervisors Andreas Runhäll and Daniel Irekvist for guiding us through this spring, holding our heads up high in times of difficulty as well as being a source of inspiration, providing large amounts of interesting ideas and viewpoints. We also want to thank Anders Eriksson and all others at Volvo Cars that have helped us by providing information and guidance. Your help has been invaluable to us.

Additionally, we want to thank our examiner Torbjörn Lundh and our opponents Ebba and Rebecca for giving us great feedback which provided new perspectives, helping us towards a better report.

Furthermore, Oscar would especially like to thank Jonatan. Johanna wants to thank David for being her rubber duck and rock, and her cat Lovis for reminding her when to take a break. Lastly, we would like to express a thank you to our fantastic friends and respective family, both of which have provided immense support in these odd times.

Oscar Forsman & Johanna Warnqvist, Gothenburg, July 2021

Contents

1
Introduction

Road traffic injuries are the leading cause of death for people aged 5-29 years and the eighth most common cause of death for all age groups worldwide [1]. Although the problem is multifaceted, increasing the overall safety of our vehicles is an important step in reducing the number of fatal accidents in traffic.

Several vehicles are equipped with passive safety features protecting the driver during an impact, such as crumple zones, airbags, and seat belts. Though, in recent years the field of active safety has gained an increasingly prominent role in further developing the safety of vehicles. Active safety functions aim to avoid critical situations before they happen, or in the case where a collision is unavoidable, reduce their severity. An example of active safety is advanced driver assistance systems such as Automatic Emergency Braking (AEB) and Forward Collision Warning (FCW) systems, both of which uses information of the traffic scenario to aid the person driving the vehicle.

Since active safety functions must operate proactively, there is a requirement to make predictions of the future based on the current available information. For the predictions to be as accurate as possible, extensive amounts of data describing the traffic situation is needed. Furthermore, due to the multitude of different scenarios a vehicle might encounter in traffic, finding appropriate models performing well in all of them is a complex task that requires careful judgement.

In this thesis, data describing routine driving behaviour is investigated for the purpose of improving an FCW system. The aim is to find possible improvements of the adaptability of the FCW system to take different drivers and their specific behaviour into greater consideration.

1.1 Background

Volvo Cars is an automotive company based in Gothenburg, Sweden, that manufactures premium cars marketed worldwide. With the aim of reaching zero collisions in traffic, Volvo Cars has developed several advanced active safety systems which are continuously improved to make their vehicles safer [2]. The FCW system on Volvo cars is part of a larger collection of subsystems dealing with accident prevention called City Safety. Together with other safety functions, these systems serve to protect the passengers of the vehicle as well as other road users.

The role of an FCW system is to aid the driver by issuing warnings when potentially hazardous situations are encountered. Through the use of camera and radar sensors the system can detect objects in front of the vehicle and assess the risk of a collision with any of the detected objects. The goal is to give an inattentive driver sufficient time to act, primarily by braking, and mitigate the damage of a collision or to avoid it completely. It is also important for the system to not alert the driver excessively in situations which they would not themselves deem to be dangerous, as this would be considered annoying, reduce driver acceptance, and potentially cause the driver to ignore the alerts entirely [3], [4]. Therefore, the timings of these warnings are crucial to their effectiveness.

On Volvo cars equipped with an FCW system, the warning consists of a sound and light signal as well as a short brake impulse to catch the driver's attention [5]. Though the time at which the warning is issued is adapted to what is currently ahead of the vehicle, there is also a need for adaptation in regard to different drivers and their specific driving style and preference. To reduce the risk of the warnings being regarded as irritating with the consequence of lessening the potential benefit of the FCW system, there is an available option for the driver to choose between three different settings to regulate the sensitivity of the system. By choosing *Early* the warnings are issued further in advance compared to *Normal*, while the *Late* option results in the warning being provided at an even later stage compared to the other two options.

The availability of just three options might be regarded as a limitation in some situations. It is reasonable to assume that a finer adjustment would be needed for every driver to perceive every FCW timing as perfect. Individual preferences might dictate that a certain option would be regarded as too early in one situation whereas in other situations the same setting might be perceived as too late. Another problem is that a new driver must actively choose to change the setting if the previous driver had another preference. These circumstances together raise the question if the adaptation could be done automatically without the need for the driver to choose an option themselves. The potential benefit of this would be that the system could be finer tuned to each specific driver while also improving the user-friendliness by not requiring the driver to choose a setting themselves.

1.2 Scope

The aim of this thesis is to investigate possible improvements of the adaptability for an FCW system based on information of previous routine driving behaviour. The goal is to find a connection between how a driver behaves in everyday driving situations and how that same driver might react in a more critical situation such as shortly before a collision or near-collision. These short events before a possible collision, where a forward collision warning would be suitable, are in this thesis referred to as *FCW events*. The choice of basing the prediction on routine driving has the potential benefit of finding a good adaptation prior to when a critical situation occurs. This thesis does not investigate connections between consecutive FCW events for the same driver, or how a driver would change their driving behaviour after collision warnings have been issued.

Driving behaviour is in this study represented by several measurements of signals collected from test vehicles and stored as a multivariate time series in a dataset. Among the available signals are measurements such as pedal position, vehicle speed, acceleration, and steering angle, as well as information related to obstacles in front of the vehicle that might be subject to triggering the FCW. All signals used are collected by sensors that are also available on production cars. The reason behind this is to not use more detailed information than what would be available to the City Safety system in a real scenario. Notably, no video recordings of the driver were collected, meaning that information such as gaze direction and other visual information about the driver's behaviour is unknown. The only information available about the driver's behaviour are through the inputs given to the vehicle through mechanical inputs.

In order to find patterns related to driving behaviour in the data, an extensive exploration of the data is made using different data analysis tools. The focus is on finding characteristics of each time series, and search for patterns among these characteristics. A search for natural groupings among different drivers is performed using clustering methods. Furthermore, correlations to specific attributes such as location, length of drive and other situational dependencies are also investigated. This thesis covers an investigation of whether routine driving can give information about how the same driver would act in a critical situation. It does not cover the actual model that would use this potential information to adapt the car.

1.3 Related work

There are multiple studies related to the functionality of FCW systems and their effect on driving behaviour. The field of driver modelling in general dates back to at least 1938 with Gibson and Crooks [6] theorizing each driver to have a specific *field of safe travel*, which road boundaries, obstacles and other drivers impose limits to. Since then, there have been multiple attempts of modelling the behavoiur of humans in various driving situations [7], [8]. As stated by Plöchl and Edelmann in 2007 [7], "... driving is a complex physical and mental individual process ...", and it is concluded that there is still no universal model useful for all applications. They maintain that there is not even a consensus on which model works best for one specific situation, although it is argued that some have been shown to work better than others in certain applications, which is to be expected.

A useful tool to use when discussing driver models is to make a distinction between the different time frames in which various phenomena occur. Michon [9] has proposed to assess driving behaviour on a strategic, maneuvering, and operational level, each associated to a different time span. The strategic level is related to high-level considerations such as route choice and mode of transport. For example, whether the driver decides to take a break along the way or perhaps take a detour. On the maneuvering level, one places behaviours such as overtaking, lane choice, and obstacle avoidance. Lastly, behaviours on an operational level are tied to a shorter time span which includes actions such as steering, braking, and other basic inputs given by the driver. While behaviour on each of these three levels might differ individually between drivers, they are also coupled such that behaviour on one level might influence the other two. For example, a driver's willingness to overtake another vehicle or not, might be decided by their actions on the operational level and tied to how much the driver is willing to steer, brake and accelerate. Similarly, there might be a connection between the driver's route choice and their likeliness to encounter certain situations.

Another important distinction relevant to this thesis, is the difference between categories of models that focus on routine driving versus others that focus on more critical scenarios. A main difference is that for near-collision models which describe some evasive maneuver, the reaction time of the driver is often assumed to be in the range of 1-2 seconds, whereas for many models describing routine driving, actions taken by the driver are assumed to be performed continuously in response to the current driving scenario, often with a very short delay [10]. The discrepancy between these two approaches and their modeled behaviour is further investigated by Markkula et al. [11] where a number of driver models were fitted to describe human truck driving in a simulated near-crash scenario. The conclusion drawn is similar to that of Plöchl and Edelmann, namely that choosing which model is best suited for a specific situation might not have a clear answer.

Moreover, braking behaviour was analysed in naturalistic rear-end crashes and near-crashes by Markkula et al. [12]. In this study, it was found that braking behaviour is strongly correlated with the urgency of the situation. More specifically, the brake reaction time for the driver depends on the visual looming, a measure of how the lead vehicle expands on the drivers retina. The suggestion made, is that the braking performed by the driver is not a response to some single onset, for example a brake light. The idea is that the brake response could rather be seen as a form of evidence accumulation, where the driver starts to brake when a certain threshold of looming is reached.

Concurrent to mathematical modelling of driving behaviour, there have also been several experimental studies on the topic of FCW. A large contribution to the field is the study from 1999 by Kiefer et al. [3] under the Crash Avoidance Metrics Partnership (CAMP) established by Ford and General Motors. FCW being an emerging technology at this time, key objectives for such a system were investigated and defined to better understand functional requirements. Possible test procedures to evaluate performance of FCW systems were also investigated in this project.

Since then, additional research has been done, as the technology has matured. The question of evaluating performance of FCW systems is especially important in order to compare different solutions and to verify their effectiveness. Studies have been made where drivers were subjected critical driving scenarios in simulation environments. A main objective has been to evaluate the difference in behaviour with, and without, an FCW system in place. An example is the investigation by Jamson et al. [4] where they also investigated the effects of adapting the FCW timing according to the individual driver's reaction time. The findings from this was that an adaptive system has the potential of being less irritating and stress-inducing.

In contrast, one should be careful when judging the results from studies based on repeated exposure of critical events as in the above. This was clearly shown by Ljung Aust et al. [13] where repeated event exposure to critical lead vehicle braking was investigated. The results from this study highlighted that analysis done by averaging over repeated events may have limited generalisability to real-world driving situations. The reason for this was argued to be that repetition of events increases drivers' expectancy and might lead to them adjusting their behaviour. In more naturalistic driving situations however, critical events usually appear more unexpectedly from a driver's perspective. This might affect their behaviour which is not captured in a simulation where events are repeated.

2

Theory

In this chapter some important concepts and methods of data analysis are described. A thorough understanding of these concepts is fundamental for evaluating the methods and to later interpret the results. In the first subsection is a description of principal component analysis which is a tool to lower the dimension of the data. Clustering and three specific clustering methods are described next, and finally three different clustering scores are described. Clustering was used in the search of natural groupings in the data set, and the scores was used to evaluate these clusters.

2.1 Principal component analysis

Principal Component Analysis (PCA) [14] transforms the current data set into a new one by creating a new coordinate system. All component vectors are mutually orthogonal and all components are uncorrelated. Further, each component vector is a linear combination of the original features.

The principal component vectors are defined as the normalized eigenvectors of the covariance matrix S, where S is defined as

$$S = \frac{1}{N-1} BB^T. \tag{2.1}$$

Here B is the standardized data set where each column is one sample, and N is the number of samples. The eigenvectors are sorted by eigenvalues such that the eigenvector with the highest eigenvalue becomes the first principal component vector, and the eigenvector with lowest eigenvalue becomes the last principal component vector. The elements of the principal component vectors acts as coefficients for the linear transform from the original features to the new basis, such that

$$Y = P^T B \tag{2.2}$$

where P is the matrix with the principal component vectors as columns. Y is the coordinates of all data points in the new coordinate system.

PCA can be used to find information about which original features contributes most when dividing the data. The elements in B are standardized and therefore the coefficients in P decides which features will be more prominent for certain components. A larger absolute value in a principal component vector indicates that the corresponding feature contributes more to the variance of that component than a feature with a lower coefficient. Analysing the coefficients in the top components will hence give information on which features are contributing more when dividing the data.

Another important tool included in PCA is dimension reduction. To obtain dimension reduction, it is possible to keep only the top components with highest eigenvalues in further analysis by either choosing a number of components, or by choosing to use the number of components required to cover a certain percentage of the total described variance. The described variance ratio of each component is equal to the eigenvalue of that component divided by the sum of eigenvalues of all components, since the sum of all eigenvalues also are the total variance of the data set.

2.2 Clustering

Clustering is a way to find natural groups in a data set and label them even when the data set does not have any ground truth labels. Points that are close or similar to each other in some sense gets grouped together with the assumptions that those points have more in common than points further from each other. This is a way to find natural patterns and groups in the data that can later be analysed to understand what the groups represent. There are many different clustering algorithms that uses different measures to decide which points are similar and how they are best divided based on this. In this section three different clustering methods are described: k-means, DBSCAN and agglomerative clustering. Not all clusters can be well found by all methods due to the differences in preferences for the different methods. Therefore it is beneficial to try different methods that might capture different cluster scenarios when working with a complex data set. k-means is a simple method that gives a good baseline for what to expect from other methods. DBSCAN has the advantage that the number of clusters are decided based on the density, and not based on a set parameter. It also has the advantage that it can label data points as outliers instead of giving all points a cluster label. Finally, agglomerative clustering has the advantage that it can use many different distance measures and techniques to decide which clusters or points that are close to each other.

2.2.1 k-means

k-means [15] is a simple and very common clustering algorithm. The algorithm takes the number of clusters k as input and tries to find the optimal k points in the same room as the data set such that each point is the mean of all points in one cluster, also called the centroid of that cluster. This is done by first initializing k centroids in some way, either by choosing k of the data points or totally at random from the same room. Each centroid gets a label associated with it. Thereafter three steps are repeated until a break condition is fulfilled.

1. All data points are assigned labels based on which centroid is closest to them.
2. The centroids are updated to be the mean of all points belonging to their cluster.

3. The distance between the new and old centroids are compared with a given threshold.

When the distance is smaller than the threshold, the algorithm has converged and will finish. Otherwise, the three steps are repeated, starting with assigning new labels to all data points based on the new centroid positions.

k-means minimizes the within-cluster sum-of-squares criterion, which is defined as

$$\sum_{i=0}^{n} \min_{\mu_j \in C} \left(\|x_i - \mu_j\|_2^2 \right) \tag{2.3}$$

where n is the number of samples, C is the set of centroids, x_i is the ith data point and $\|\cdot\|_2$ is the Euclidean distance. Another word for within-cluster sum-of-squares is inertia. To minimize this equation, the centroids will be chosen such that all data points are as close as possible to some centroid and each data point will belong to the cluster of the nearest centroid.

The k-means algorithm will always converge but might get stuck in local minima. The result depends only on the initialization, and therefore it is common to do several runs with different initializations and take the result that minimizes Equation 2.3.

2.2.2 Agglomerative clustering

In Hierarchical clustering [15], a tree structure is used where each node represents a combination of samples that creates a cluster. The leaves of the tree are all clusters containing only one sample. The root of the tree is on the other hand the cluster containing all samples. All other nodes are different combinations of samples, containing the union of all samples that are included in the children of the node, and a subset of the samples contained in the parent node.

In agglomerative clustering, a bottom-top approach is used when constructing the tree for a given data set. All samples start in different clusters as root nodes of their own subtree. At each step two root nodes representing different clusters are joined together with a parent node representing the union of the two clusters. The new node becomes the root node of this new subtree containing the union of the two joined subtrees. Which subtrees are joined together are the ones minimizing some linkage criteria or cost function. The joining continues until all subtrees have been joined together as one tree.

The choice of linkage criteria for agglomerative clustering varies. One possibility is to use the Ward linkage, where the sum of squared differences within all clusters are minimized. This way, each cluster will have minimal variance. The three other common approaches instead looks at which pair of clusters minimizes a criteria and should therefore be the two clusters that are merged. This allows the two clusters that on average are closest to each other to merge, but what is seen as closest is different. The Average linkage chooses the two clusters for which the average distances between all samples in the two different clusters are minimized. The Complete linkage minimizes the maximum distance between two samples in different clusters, ignoring all other samples in the clusters. The Single linkage is somewhat the opposite. It chooses the two clusters for which the minimal distance between two samples in the different clusters are minimized. The Ward linkage requires Euclidean distance to be used, while the other three are open for other distance measures as well.

To decide which clusters to use in the end, there are two ways. Either a number of clusters are chosen and the algorithm is stopped when there are this many subtrees. The root nodes of these subtrees will describe the different clusters. Another approach is to choose a threshold for the linkage criteria and stop merging when no clusters has a distance below this threshold.

2.2.3 DBSCAN

DBSCAN [15] stands for Density-Based Spatial Clustering of Applications with Noise. It is a method that finds clusters of similar density with the ability to set points in non-dense areas as outliers. The number of clusters are not a set parameter, instead the algorithm will decide the number of clusters on its own based on the given parameters. DBSCAN has only two parameters: the minimal number of samples k and the maximal distance ϵ. Each point is considered a core point if it has at least k samples, including itself, within a range of ϵ. Any distance measure can be used to determine the distances between points.

When deciding labels for all data points, the algorithm goes through all unlabeled points in order. When it finds a point p that has at least k points within ϵ distance, this point p is a core point and becomes the first point in a new cluster. All points within ϵ distance from point p are also added to the cluster. If they are core points as well, their close neighbours will also be added to the cluster and this continues iteratively until no more points can be added. The algorithm continues to search for core points among the unlabeled data and may find several clusters where each will consist of at least k points. When all data points are checked, the data points without label will be set as outliers.

2.3 Clustering scores

With a labeled data set it is possible to compare the cluster labels with the ground truth labels. This is not the case with unlabeled data. Instead other measures must be considered when deciding whether a specific clustering is good or not. The clustering methods have different ways to decide what is the best clustering given the parameters and method used, but to compare different parameters and even different clustering methods, more general clustering scores are needed. In this section three different clustering scores for unlabeled data are described. In general, they check if the points in the same cluster are close to each other, and that the points in different clusters are far from each other, but this is done in different ways.

2.3.1 Silhouette score

The silhouette coefficient for each sample i is calculated from the mean intra-cluster distance (a_i), which is the mean distance to other samples in the same cluster as sample i, and the mean nearest-cluster distance (b_i), which is the mean distance from sample i to all samples in the nearest other cluster [15]. Any distance measure can be used to calculate these distances. Using the calculated values of a_i and b_i, the silhouette coefficient s_i for sample i is obtained from

$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)}. \tag{2.4}$$

The coefficient can take values between -1 and 1 where close to 1 is a good clustering with dense clusters and far between different clusters, while a value below 0 indicates that sample i might belong to the wrong cluster. To get a silhouette score for the whole set, it is common to use the mean of all silhouette coefficients for different data points.

2.3.2 Calinski-Harabasz score

Calinski-Harabasz score is also called Variance Ratio Criterion, due to it being defined as the ratio of the between-clusters dispersion mean and the within-cluster dispersion [15].

Let E be the set of data points with n_E points and k be the number of clusters the points are clustered into. If C_q is the set of points in cluster q, c_q and c_E the center of cluster q and the data set E respectively, and n_q the number of points in the cluster q. Then the between-clusters dispersion matrix, B_k, is defined as

$$B_k = \sum_{q=1}^{k} n_q (c_q - c_E)(c_q - c_E)^T \tag{2.5}$$

while the within-cluster dispersion matrix, W_k, is defined as

$$W_k = \sum_{q=1}^{k} \sum_{x \in C_q} (x - c_q)(x - c_q)^T. \tag{2.6}$$

Using these matrices the Calinski-Harabasz score s is defined as

$$s = \frac{tr(B_k)}{tr(W_k)} \cdot \frac{n_E - k}{k - 1}. \tag{2.7}$$

Higher between-clusters dispersion and lower within-cluster dispersion indicates that the data set is well clustered, and will give a high value of s. No maximum value exists for this value. All values in W_k and B_k are positive due to the square in their definitions, see equation 2.5 and 2.6. The number of clusters must be positive and never higher than the number of data points to be feasible. This means that the value of s never goes below 0.

2.3.3 Davies-Bouldin score

The Davies-Bouldin score [15] measures the average similarity between clusters. In practise, this is a comparison between the distance between clusters and the size of the clusters. It compares each cluster i with the most similar other cluster, j. For each cluster the cluster diameter s_i is calculated with Euclidean distance,

$$s_i = \frac{1}{n_i} \sum_{x \in V_i} \|x - c_i\|_2 \tag{2.8}$$

where V_i is the set of all points in cluster i, c_i is the center point of cluster i and n_i is the number of points in the cluster. Additionally, for each pair of clusters, the distance between the cluster centroids is calculated such that

$$d_{ij} = \|c_i - c_j\|_2.$$

Together this gives a similarity for each pair of clusters,

$$R_{ij} = \frac{s_i + s_j}{d_{ij}}$$

and the Davies-Bouldin score, s, is finally defined as

$$s = \frac{1}{k} \sum_{i=1}^{k} \max_{i \neq j} R_{ij}$$

where k is the number of clusters. The perfect score is 0 and lower scores in general indicates a better clustering than higher scores.

3

Methods

In this chapter the method is described of how to investigate how the behavior in routine driving scenarios is mirrored in the behaviour of the driver in a near-collision event. The data set is processed and analysed in different ways to extract as much information as possible about the correlation between the behaviour at routine and critical driving situations. The chapter is divided in three different sections. The first section decribes the data set, the second section describes the preprocessing steps and the last section describes the different data analysis methods that was applied to the processed data set.

3.1 Description of data

The data set used in this thesis consists of data collected during development of the Advanced Driver Assistance Systems (ADAS). The data consists of many different signals combined into multivariate time series. Collection of data is done using many different subsystems, and different subsystems have different standards of how often the value for the signal is collected. Therefore there are different frequencies of time stamps for different signals.

The signals are measurable things from the car, such as velocity, pedal movements, steering angles and wiper status. In addition there exists detailed information about the vehicle in front, the object data, which describes if any vehicle is detected in front of the own car and many details about this potential target. The information in the signals are represented in three different ways. Some are discrete where each number represents a different state such as if the brake pedal is pressed or not, or how often the wipers are set to move. Some signals are fully continuous. Velocity belongs to this category. The third category is signals that are sometimes continuous, but also has a default value if no real value exists for the moment. The object data is included here. If no target exists the default value will be used, but while a target does exist some values such as the velocity of or distance to the object will be continuous.

The object data is the main reason to use this data set when looking at forward collision related topics, but the extent of other signals that could be of interest when analysing a drive live is also a contributing factor. By working with the signals that are always available in the cars, the utility of the findings of this thesis increases since they can more easily be implemented in real cars. Another perk of this data set is that it contains very many hours of driving, and all is done in real traffic scenarios rather than on a test track.

For this specific research question, a downside of the data set is that it is collected by only a few drivers and a few cars, which decreases the generalisability of the results when the driver behaviour is in focus. It is also not labeled with which driver drove which drive, only the car used. On the other hand, the data is collected in different parts of the world, which increases the generalisability.

3.2 Preprocessing

Preprocessing, including data cleaning and feature engineering, is necessary to be able to handle the large amount of information contained in each time series. A motivation for why feature engineering is necessary instead of analysing the time series as they are is given in the first subsection. Thereafter comes a description of the different steps from having the full time series to having a manageable data set that can be used for further analysis. Before this is done, the data is split in two sets, one with all drives without any FCW event called the no-FCW set, and one with all drives that has at least one FCW event called the FCW set. Fitting parameters for different preprocessing steps will be done using the set without FCW events, and the same parameters will be used when processing the other set.

Only a handful of the signals available in the data set will be used. The data set consists of very many signals where some are very correlated and many are unrelevant for this investigation of behaviour. A set of 33 signals was chosen for this investigation. These signals are believed to cover enough information about the actions of the driver, the car and the object in front that would be relevant in a comparison of routine driving and near-collision events.

Before any other step can be made, the signals of each time series are interpolated to have the same time stamps for all signals. A time step of 0.025 seconds was chosen and at each time step each signal was assigned its latest registered value. For this to work, the time series was set to start where all signals had registered at least one value. Extrapolation is much more unreliable than interpolation. Therefore no extrapolation was used, and the time series was ended where the first signal had its last value.

3.2.1 Why feature engineering is necessary

When analysing data it is often necessary to have a measure of how similar two different samples are. This is usually done by using a distance measure and say that samples that have a small distance between each other are more similar than samples with large distance between them. Distance between time series can be calculated in different ways. The two most common methods are Dynamical Time Warping (DTW) and Euclidean distance [16].

Euclidean distance is a continuation of the Euclidean distance for two vectors, but here the values at each time step in one time series are compared with the values at the same time step in another time series. The total Euclidean distance between two time series will be the sum of the Euclidean distances for each time step. This requires the time series to be of the same length, which makes it impossible to compare two drives of different length. This could be worked around by using windows of a set length, but another problem remains. Comparing two identical events that starts at different time steps in the two time series can give a very large distance although they would be identical if aligned. Working with driving data it would be preferable if these events where found to be equal.

Dynamic Time Warping [17], [18] does not require the time series to be of the same length. The method can also handle events that are slightly shifted, due to the methods ability to shift and squeeze sequences so that two time series could fit better if only the speed and start of the events differ. However, one final problem is present in both methods. DTW may be able to shift and squeeze, but it is not able to change the order of events in a time series.

In the driving data used for this thesis, each drive consists of many different, natural driving scenarios in different settings. If two different time series contain data from the same driver, but the first series contains event A and B, and the second series contains the same events but in opposite order, it would be preferable if the time series are seen as quite similar. This is not the case for any of the common distance measures for time series.

Instead of comparing the time series in their original form, it is possible to extract information from a time series and save this as a single data point, caring less about the order and more about characteristics. This allows simpler analysing tools to be used when analysing the data since the data no longer consist of time series but of ordinary multidimensional data points. How this is done in a representable way is described in the following subsections.

3.2.2 Tsfresh – from time series to data points

Tsfresh [19] is a Python package that transforms a single or multivariate time series of any length into a single data point by extracting features from the time series. Tsfresh uses hundreds of different functions and apply these to each signal that the time series is built from. Sometimes a function requires some parameter to be set, and in this case the function will be applied to each signal several times, one time for each parameter value. It is the combination of signal, function and parameters that is seen as one feature. For example, the functions can be the mean of all values in the time series, the number of times the value goes over a certain threshold or the number of peaks. Combined with certain signals such as velocity or amount of braking, a feature can be the mean velocity or the amount of time the brake pedal is pressed more than 50 percent. Another feature would describe the amount of time the brake pedal is pressed more than 20 percent, where it is the same signal and function as before, but another parameter value. All values obtained from all different combinations of functions, signals and parameters will be combined as the coordinates for the new data point that is constructed to represent the full time series. The tsfresh package has been well tested and thanks to its many different functions it represents time series well. The data points can later be used together with ordinary data analysis tools intended for such data instead of considering special methods created for time series.

For this thesis, the time series are of very different lengths due to the drives being of different length. To make the data more comparable, a window of five minutes is applied to all data such that the time series are cut in five minute pieces. The input to tsfresh becomes these shorter time series windows, and the output is one data point for each window.

3.2.3 Data cleaning

Some features might result in NaN-values, if the function is not well defined for the time series data. After running tsfresh, all features which has a NaN value for any data point are therefore removed to maintain the same dimension space for all data points. All features that are constant for all files are also removed, since they do not contribute with any new information. If kept they could cause problems if they would contain floating errors that blow up when applying standardization later on. A few other features was removed due to not being well suited for this kind of data, and contributing too much to the high number of dimensions. These steps helps to reduce the dimension of the data points to something more manageable.

Furthermore, some data points were also excluded. The data points corresponding to the last window of each drive are removed since these windows have different lengths than all other windows. All data points corresponding to windows where no primary target was detected during the full window is also excluded. Finally all data points from a drive containing a specific outlier is removed, since this outlier indicates that something might be wrong with the data collection from this drive.

3.2.4 Principal component analysis

The functions used by tsfresh give very different range of values for different functions. Therefore it is important to standardize the data to make all features more comparable. This is done by subtracting the mean from each feature and divide by its variance. This way all features will have mean 0 and variance 1.

Due to the curse of dimensionality, it is beneficial to reduce the number of dimensions before applying any analysis methods on the data. A very common way to do this is by using principal component analysis. To minimize the number of components and still use a representable part of the total explained variance, the number of components kept for later analysis is chosen based on the variance plot, shown in Section 4.1.

3.2.5 Apply processing on the FCW set

When the preprocessing of the no-FCW set is finished, the same preprocessing is applied to the FCW set, using the same mean, variation, feature choice and principal components as was decided for the no-FCW set. If any NaN value remains in the data after all preprocessing is done, these data points are removed with the motivation that they must be outliers or faulted in some way when they give error where no sample in the no-FCW set gave error.

3.3 Data analysis

When the data transformation is done, the data can be analysed using different data analysis tools. First the components are analysed to understand their meaning and decide how much the dimension can be reduced before continuing with other analysis. Thereafter, clustering is applied in different forms and fitted to the data set to search for natural groups in the data and correlations in behaviour at routine and critical driving situations. Lastly the correlation of data points corresponding to windows in the same drive or of the same order is investigated.

3.3.1 Analysing components

The variance of different components are investigated to understand which components contribute most to the distribution of data points. Lists of the top features for the top components are analysed, and plots that show the distribution of points for two components at a time are inspected to find how the components work together.

To understand what the different principal components stand for it is possible to look at the coefficients for each feature in each component. A larger absolute value of a coefficients means a bigger impact from that feature on that component. By investigating which features have the highest coefficients for each component, it is possible to translate what each component says about the data and get a better understanding of what the shape of the data set tells.

3.3.2 Fitting the clustering methods

Three different clustering methods were applied to the data using the Python package sklearn [20]. The motivation to use clustering methods was to investigate if any natural groups are represented in the data. These natural groups could be of any form, and by using three different clustering methods, the chance of finding the natural grouping increases. Here the focus lay on how the clustering methods were fitted to the data set and which parameters were investigated. A detailed description of the clustering methods and what these parameters affect are presented in Section 2.2.

3.3.2.1 k-means

k-means finds clusters of similar variance, given a few parameters. A description of the method are presented in Section 2.2.1. The method was implemented using the sklearn implementation of k-means. See the documentation [21] for information about the implementation and the different parameters. A grid search was used to decide which parameter values were best suited for the data set, using the values in Table 3.1. The three clustering scores Silhouette score, Calinski-Harabasz score and

Davies-Bouldin score was used to determine which parameters were best suited for this data set. These three scores were optimized separately, which means that for example the parameters for optimal Silhouette score might not be the same as the parameters for optimal Calinski-Harabasz.

Table 3.1: Parameters tested for k-means. max_iter is the maximal number of iterations for each initialization and n_init is the number of initializations. More details about the different parameters can be found in the documentation [21].

max_iter	500, 1000, 2000
n_init	20, 40, 60
algorithm	Full
n_clusters	2, 3, 4, 5, 6, 7, 8, 9, 10

3.3.2.2 Agglomerative clustering

Agglomerative clustering requires either a threshold or a number of clusters to be chosen to decide when to stop the algorithm. Since a higher value of clusters are not of interest, a parameter grid search was made to see which parameters are best suitable for the data set used, including testing different numbers of clusters. To determine which parameters are best suited for this data set, the three different clustering scores Silhouette, Calinski-Harabasz and Davies-Bouldin was used. As for k-means, these scores were optimized separately and could give different sets of optimal parameters. For further information about the implementation and possible parameters, see the documentation from sklearn [22]. Table 3.2 shows the parameters included in the parameter grid search when applying agglomerative clustering on the data points without any FCW events.

Table 3.2: Parameters tested for Agglomerative Clustering

Linkage	Ward, Average, Single, Complete
Affinity	Euclidean, Manhattan
n_clusters	2, 3, 4, 5, 6, 7, 8, 9, 10

3.3.2.3 DBSCAN

Due to DBSCAN's ability to find only one cluster, no parameter grid search was used for this method. The clustering scores used for k-means and agglomerative clustering are not well suited for a clustering where some data points are labeled as outliers. They also do not work for only one cluster, hence ignoring the outliers is not enough. Comparing different parameters for DBSCAN would require some kind of score of which is best, but since this is not available the parameters was chosen in a different way.

With a set value for the minimal number of samples k, it is possible to find a suitable value for the radius ϵ. The value of k was set to be two times the dimension of the samples. Using this value the following steps were needed to set the value for ϵ.

First, for each point, the distance to the kth nearest neighbour was calculated, with the point itself counting as the first nearest neighbour. These distances were sorted in ascending order and plotted. From this plot the value of ϵ can be chosen by the elbow method. This means that ϵ is the value where the derivative of the ascending distances abruptly changes. The idea is that data points with further distance to their kth nearest neighbour should not be seen as core points but rather be in the fringe of a cluster or be marked as outliers since they are in areas of distinctly different density.

3.3.3 Classification and validation on the FCW set

After clustering on the data points from the no-FCW data set, the data points from the FCW data set was classified by using k-Nearest Neighbours (kNN) with $k = 1$. This means that for each data point p in the FCW set, the closest neighbour from the no-FCW set was found, and the label of this neighbour became the label of data point p as well. For a higher value of k, the label will be chosen as the most common label among the k nearest neighbours. $k = 1$ was chosen with the motivation that the data set is very large. The clustering scores for the labeled FCW set are calculated to compare with the scores from the no-FCW set. If both scores are similar and good this means that the clusters exists in both sets and the result is generalisable. If the result is much worse for the FCW set, this means that the clustering is overfitted on the no-FCW set and not generalisable to other drives.

3.3.4 Comparing clustering of data points and time series

Since an FCW event only make up a very small part of the full window the data point represent, a further investigation of clustering and the correlation of routine and critical driving behaviour was made. Time series containing 5 seconds before and after the first FCW event in each time series was extracted.

First the event time series was given the label of the data point corresponding to the window where the event occured, after these data points had been classified using kNN as described in the previous subsection. Silhouette score using DTW as distance measure was used to find how well clustered the events were using these cluster labels.

Clustering was also applied directly on the event time series. k-means could not be used since it is not suitable for time series and uses Euclidean distance by definition. Agglomerative clustering was used following the same procedure as explained in Section 3.3.2.2, but with DTW as affinity, and Ward linkage which requires Euclidean distance was excluded. DBSCAN was also used with DTW as distance measure. Only Silhouette score allows other distance measures than Euclidean, and hence this was the only score used for the time series.

After clustering the event time series, the FCW set was classified. All windows from the same drive was given the same label, decided by the label of the first event in that drive. Thereafter kNN with $k = 1$ was used to classify the data points in the no-FCW set.

3.3.5 Correlation of windows

The correlations of different windows, either with same order or from the same file, was investigated. Plots showing where the first and last window for each file are located compared to other windows show if starting and stopping give any unwanted or unexpected results. The correlation regarding in which part of the world the data was collected was also explored.

4

Results

The results are presented in this chapter, divided in different sections based on topic. First the variance of different components are presented and a choice is made of how many components will be used for the other analysis steps. The information contained in the top components are investigated in the next section. Thereafter the clustering results are presented, first the results from clustering on the data points in the no-FCW set, and after that the results from when clustering on the event time series. The last section shows results regarding sensitivity to situations.

4.1 Eigenvalues for principal components

In order to reduce the dimension the variance of different components are investigated. In Figure 4.1 two different plots are shown. The left plot shows the variance of the different principal component vectors sorted in descending order. The variance for the first component is large, but the variance decreases quickly for the first components and stabilizes at a quite low value where it is almost constant in comparison.

In the right plot however, the cumulative explained variance ratio is plotted. This shows that the first ten components make up a third of the total explained variance. The cumulative sum is flattened for higher number of components and becomes almost linear, which indicates even further that the variance become close to constant for later components.

These plots together indicate that the remaining components consist of noise with low variance, but due to the high dimension they still make up a large part of the total variance. For continued analysis 60% of the total explained variance ratio, and hence the first 53 components, was decided upon to use as a golden road between keeping enough information and trying to exclude as much noise as possible.

4.2 Interpretation of components

In Figure 4.2 the data points are plotted for four different component planes. These plots show the data points divided in three different categories. The two sets with and without FCW events in the drives are shown, but the FCW set is divided in two parts. One for all data points corresponding to windows without any actual FCW

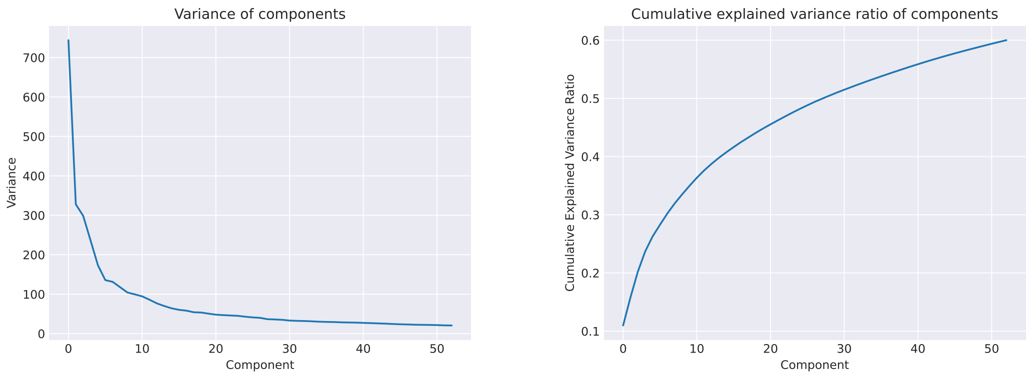


Figure 4.1: Variance of different principal components. The variance rapidly decreases for the first components and stabilizes at a low variance. The same behaviour is visible in the cumulative explained variance ratio, where the first 10 components make up a third of the total variance, while 53 components make up 60%.

event, and one for those with at least one event in their window. The plots show that the distribution for these three sets are very similar. The difference in variance is only a consequence of the difference in size of the sets. It also noticable that no distinct clusters can be seen in the first components, except for component 5 and 6 where a few data points builds a cluster seperate from all other points.

The coefficients for each of the top eight components were analysed to see which features contribute the most to each component. An interpretation was made based on these top features for each component, to see what each component might explain. The result is showed in compact form in Table 4.1 and further elaborated here.

In component 0 a high value indicates a high complexity of the features describing the presence of a primary target and also a high variance in the velocity of the own car. This means in practise that if the scenario switches often between having a primary target and not having a primary target in the time window the data point represents, than component 0 will have a high value. If only a few changes between these two states occur, the value will be low.

Component 1 has high value if the values related to the primary target is different in the start and end of the time window and is therefore not of much interest for this type of data where the window of each data point often contains several different scenarios in different orders.

The next three components are easier to interpret. A high value in component 2 indicates that steering is a prominent part of the window, and a high value of component 3 means high velocity while a low value indicates that the driver hits the brake. The next component describes the weather. A high value for this component indicates that it is raining or snowing.

Component 5 and 6 are related to the country in which the data is collected, because different countries uses different units of speed. This is visible in Figure 4.3 where the distribution of data collected in the US has a tail of higher values for both

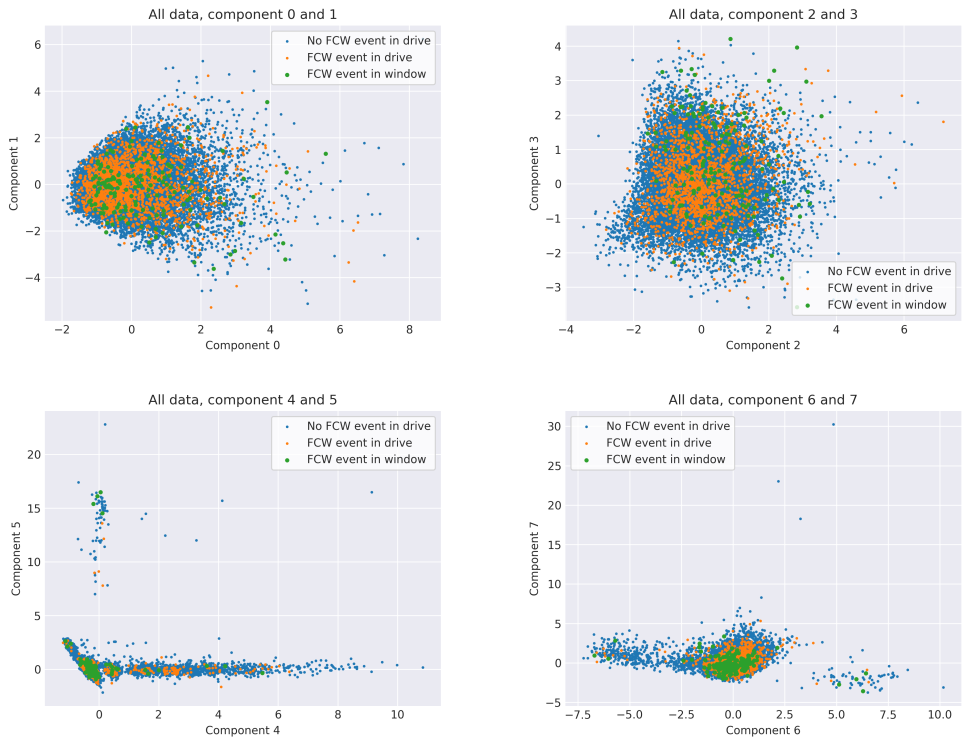


Figure 4.2: Distribution of data for different component pairs. The blue points are data points from the set of drives without any FCW events. The yellow and green points are both data points from the set with drives that has at least one FCW event. Yellow are the data points in this set that corresponds to windows where no event occured. Green are the data points corresponding to windows with an actual event. Note that none of the plots show any clear separation between different types of windows. This was true for all investigated components.

components, while the data collected in EU is more concentrated. This has to do with the representation of the information. The signal that is used here gives information of the unit of the speed limit. It has a default value which is km/h, but when it sees a speed limit sign, it detects which unit should be used for this sign. This means that in EU this value never changes since the real value is the same as the default value. In US however, the default and non-default value is different and the signal will switch between these values when it sees a sign or goes back to the default value. Therefore the variance for the drives in US is higher than the variance for drives in EU. The components are also correlated with usage of the Active Cruise Control function which explains why there is still some variance for the drives in EU. The difference between the components are hard to distinguish when looking at the coefficients.

Finally, the last component analysed further is component 7. Similar to component 0 this component seems to have higher values when the situation varies between having and not having a primary target. However, difference in acceleration does not seem to have a big impact here, instead a decrease in velocity contributes. This can be interpreted as component 7 has high values when the car is driving on a

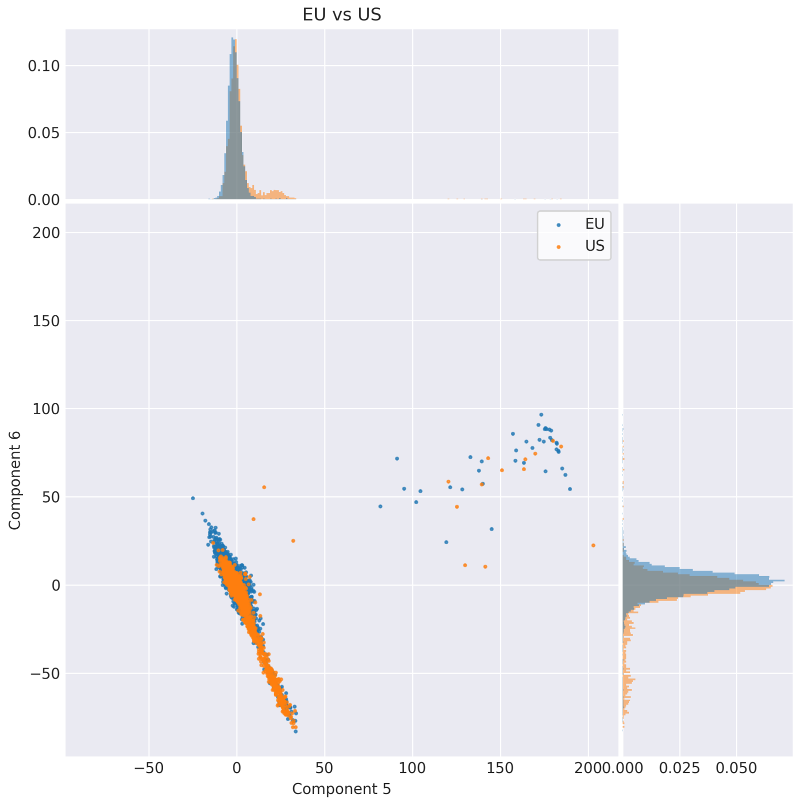


Figure 4.3: Distribution of data collected in US and EU respectively shown for component 5 and 6. Note that for both components the data collected in US has a tail of higher values while the data collected in EU is more concentrated. This is clearly visable in the two histograms describing the distribution of points. The reason for this difference, is tied to how the information about the country in this specific signal is stored.

country road and follows another car, while component 0 has high values for cars that are driving in cities with cars in front that often slows down for different reasons as happens in city traffic.

4.3 Clustering windows of routine driving and validating on event data

The results from applying different clustering methods to the data points is presented in this section. The optimal parameters and corresponding scores are presented for the different clustering methods k-means, agglomerative clustering and DBSCAN. The scores for the different methods are compared, and the result is validated on the drives containing FCW events and also on short time series around the specific events.

Table 4.1: Interpretation of the top eigth components based on their coefficients. Data points corresponding to windows fitting the description will have a high value for that component. The last column gives some examples of important features for each component. These features will have high valued coefficients.

Nr	Description	Feautures with high coefficients
0	Switching between having and not having a vehicle in front. High variance in velocity and acceleration.	Standard deviation of time to collision. Longitudinal position of primary target. Ratio of required negative acceleration, and ratio of acceleration, above six times the standard deviation.
1	Having an object in front either in start or end of window, not both.	Linear trend of object data
2	High amount of steering.	Mean absolute change of Pinion Steer Angle. Standard Deviation of Yaw Rate.
3	High velocity and small amount of braking.	Mean, median and sum of values for velocity. Negative mean, negative median and negative sum of values for when the brake pedal is pressed.
4	It is raining or snowing.	Maximum speed of wipers. Mean of wiper speed.
5	Car is driving in US and partly uses Active Cruise Control.	Features relating to activation of Active Cruise Control and having miles per hour as speed unit.
6	Car is not driving in US and partly uses Active Cruise Control.	Features relating to activation of Active Cruise Control and not having miles per hour as speed unit.
7	Switching between having and not having a vehicle in front. Decreasing velocity.	Negative autocorrelation of different signals connected to the primary target. Kurtosis of velocity.

4.3.1 Parameters and results for k-means

The parameter search for k-means gave the results in Table 4.2. The Silhouette score and Calinski-Harabasz score were both highest for 2 clusters using 20 different initializations to average over. They differ though in number of iterations each run wanted to use. While Calinski-Harabasz preferred 500 iterations, Silhouette preferred 2000 iterations. The Davies-Bouldin score also chose 2000 iterations, but the number of initializations was set to 40 and the number of clusters to 3 for the highest score.

Table 4.2: Parameters for k-means when optimized for the three different scores separately.

	max_iter	n_init	n_clusters	Score
Silhouette	2000	20	2	0.183030
Calinski-Harabasz	500	20	2	297.278
Davies-Bouldin	2000	40	3	2.43942

4.3.2 Parameters and results for agglomerative clustering

For agglomerative clustering, all scores agree that Euclidean is the best affinity measure and that two clusters is the best choice, as seen in Table 4.3. The linkage however differs. While both Silhouette and Davies-Bouldin give highest values when the Average linkage is used, Calinski-Harabasz gives higher value when using the Ward linkage.

Table 4.3: Parameters for agglomerative clustering that gives the best score for each clustering score.

	linkage	affinity	n_clusters
Silhouette	Average	Euclidean	2
Calinski-Harabasz	Ward	Euclidean	2
Davies-Bouldin	Average	Euclidean	2

The Average linkage gives a cluster containing only one datapoint while all other samples are one big cluster. For the Ward linkage, the smaller cluster has a few more points and consist of the small group of outliers in component 5 excluding the point with very different value in component 4, see Figure 4.4.

4.3.3 Parameters and results for DBSCAN

For DBSCAN the minimal number of samples k was chosen to be two times the dimension, which gives $k = 2 \cdot 53 = 106$. The range ϵ was obtained by the elbow method as described in Section 3.3.2.3 and is chosen to $\epsilon = 12$. See Figure 4.5 where the distance to the kth nearest neighbour for each data point is plotted in ascending order. The value of ϵ was chosen as the point where the curve changes most drastically and has the best chance of avoiding setting outliers as core points. Running DBSCAN with these parameters generates only one cluster with some outliers.

4.3.4 Comparison of results for different clustering methods

The best scores achieved from k-means and agglomerative clustering are shown in Table 4.4. Since DBSCAN only finds one cluster, the different clustering scores are not applicable.

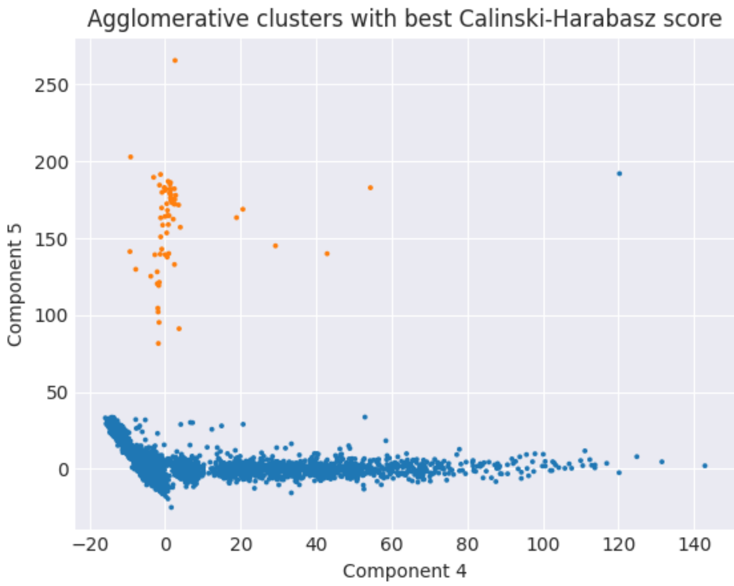


Figure 4.4: The clusters from agglomerative clustering with Ward linkage, Euclidean affinity and 2 clusters, in the plane of component 4 and 5.

The results obtained from clustering the data points in the no-FCW set with only routine driving indicates that no natural groups can be found in the data set. A good result would be a Silhouette score close to 1, a Davies-Bouldin score close to 0 and a Calinski-Harabasz score close to infinity. The scores obtained from k-means are far from optimal. For agglomerative clustering the Silhouette score and Davies-Bouldin score are much better than for k-means when optimising the parameters for these scores, but these parameters give two clusters where one of them contains only one data point. This further indicates that the algorithm would choose to have only one cluster if it was allowed to, and only finds clusters because it is forced to, not because these are natural groups of interest in the data. That DBSCAN only finds one cluster strengthens this statement.

Table 4.4: Best clustering scores when optimizing for each method and each clustering score separately. DBSCAN gives one cluster and the scores are hence not applicable.

	k-means	Agglomerative	DBSCAN
Silhouette	0.183030	0.915645	-
Calinski-Harabasz	297.278	287.999	-
Davies-Bouldin	2.43942	0.058359	-

4.3.5 Classification of and validation on the FCW set

The FCW set was used to validate the results of the clustering of the no-FCW set with routine driving. The classification method kNN with $k = 1$ was used to decide which cluster each new data point in the FCW set should belong to, based on the

Figure 4.5: Distance to the kth nearest neighbour of each datapoint, sorted in ascending order. $k = 106$ which is two times the dimension. Epsilon is chosen by the elbow method, which gives $\epsilon = 12$.

clustering labels for the data points in the no-FCW set. These labels were different depending on which clustering method was used and which score was optimized when deciding the parameters. For k-means and agglomerative clustering this gives three different scenarios each, one for each clustering score. After classifying the FCW set, the same score was calculated for the labeled FCW set as was optimized for the corresponding no-FCW clustering. The results are presented in Table 4.5. Note that the scores without value in Table 4.5 is based on the clustering where one cluster only contained one data point and the rest of the points belonged to the other cluster. No data point in the FCW set was closer to this outlier than to any other data point and hence all from this set was labeled as the same cluster.

Comparing the scores for the FCW set with the scores for the no-FCW set, it is clear that the results are worse for this set than for the routine driving. Most noticable is that the Calinski-Harabasz score is much lower for this set for both clustering methods. This indicates that the clusters found in the no-FCW set do not generalise well to other drives.

Table 4.5: Scores for the FCW set, when classifying the data based on clustering on no-FCW set optimized for the corresponding score. No value means that all data points in the set belongs to the same cluster.

	k-means	Agglomerative
Silhouette	0.112472	-
Calinski-Harabasz	21.1538	57.9246
Davies-Bouldin	3.06631	-

4.3.6 Classification of FCW event time series

A final validation of the clustering on the routine driving data was done on the time series around the events, as described in Section 3.3.4. Table 4.6 shows the Silhouette score when using DTW distance for these event time series. The label of each event is assumed to be the label that was set for the data point corresponding to the window in which the event is contained. Note that the Silhouette score is close to zero for all combinations where more than one cluster is represented. Hence there is no pattern in the event time series that corresponds to the given clustering labels.

Table 4.6: Silhouette score for event time series, based on clustering indices decided by clustering the no-FCW set, classifying the FCW set and classifying the events based on which window they are contained in. The columns represent which clustering method was used, and each row represent which score was optimised when choosing parameters for the clustering method. All scores presented are Silhouette scores since Calinski-Harabasz and Davies-Boulding not are applicable on time series.

	Agglomerative	k-means
Silhouette optimized	-	-0.00168688
Calinski-Harabasz optimized	0.01159931	0.00766757
Davies-Bouldin optimized	-	0.00149114

4.4 Clustering event time series and validating on data points

The results from the opposite direction of clustering is presented in this section. Only Silhouette score is applicable for time series since it is the only score allowing DTW as distance measure. Since k-means requires Euclidean distance, only agglomerative clustering and DBSCAN are used.

4.4.1 Parameters and results for agglomerative clustering and DBSCAN

For agglomerative clustering, the parameters giving highest Silhouette score was to use Average linkage and two clusters. This gave a score of 0.415672. The clusters consist of 74 and 92 events respectively which is much more evenly distributed than the clusters obtained when clustering on the data points.

For DBSCAN, when setting $k = 66$ since that is two times the dimension, the elbow method gives $\epsilon = 2550$ as seen in Figure 4.6. These parameters give only one cluster with 93 events and 75 outliers.

Figure 4.6: DTW distance to the kth nearest neighbour of each datapoint, sorted in ascending order. $k = 33$ and 66 respectively which is one and two times the dimension. Epsilon is chosen by the elbow method, which gives $\epsilon = 12$.

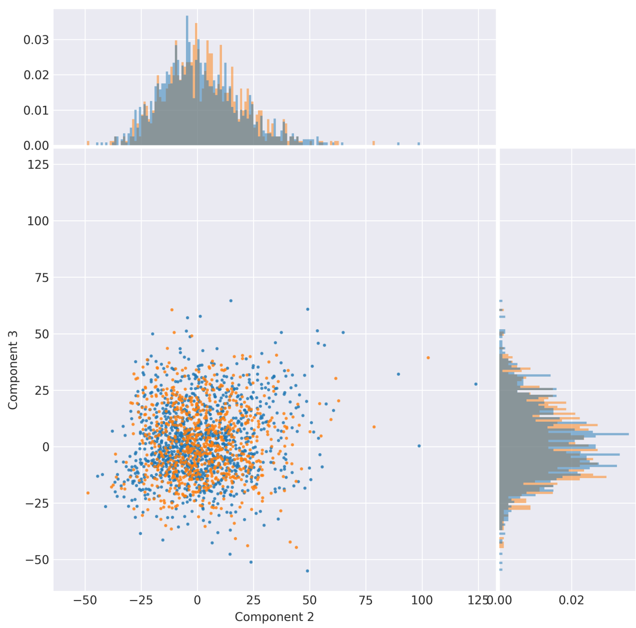


Figure 4.7: Distribution of the two clusters obtained when classifying drives with FCW events based on the agglomerative clustering of the event time series.

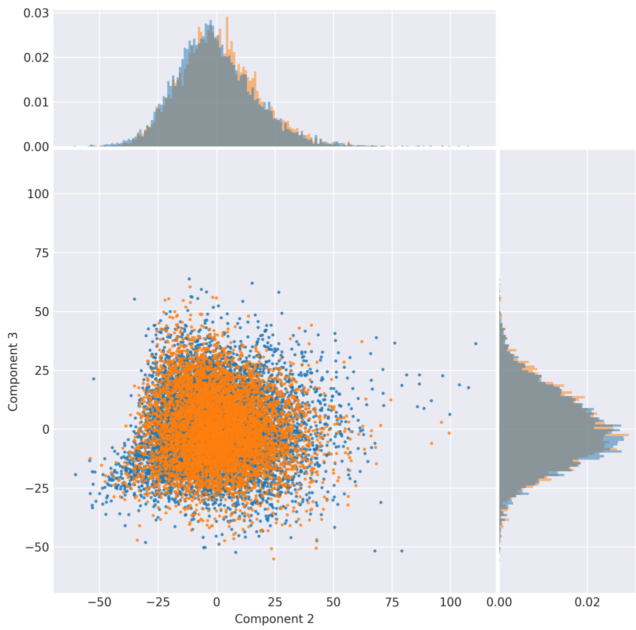


Figure 4.8: Distribution of the two clusters obtained when classifying drives without FCW events based on the agglomerative clustering of the event time series.

4.4.2 Classification of drives including FCW events

In the FCW set, all data points from the same drive was given the same label, decided by the label of the first event in that drive. Euclidean distance was used to determine the distance between the data points. With this procedure, the Silhouette score for the FCW set became 0.00415998. A Silhouette score close to zero indicates that the clusters are indistinct, and this is also visible for all the top components. In Figure 4.7 the first two components are shown as example. Note that both the yellow and the blue cluster has approximately the same distribution. This is true for all the explored components.

4.4.3 Classification of drives with no event

The data points in the no-FCW set were labeled using kNN with $k = 1$. For each point, the closest neighbour from the labeled FCW set was found, and the label copied. Euclidean distance was used to determine distances between data points and the Silhouette score obtained for the no-FCW set was 0.008056. The score is slightly higher than the score for the drives with an FCW event, but it is still close to zero. This indicates that the clusters are indistinct for this set as well, which is also visible for all the top components. In Figure 4.8 the first two components are shown as example. The distribution of the two clusters are approximately the same and hence the clusters not well seperated. This is true for all the explored components.

4.5 Sensitivity of situations

Shown in Figure 4.9 is the distribution of datapoints with and without events when not doing any cleaning of data except deleting features that are constant or contain NaNs. Note that the data is seperated in two clusters, where the left cluster is much more dense. All three kinds of windows are represented in both clusters, but only a few of the windows containing an event is in the left cluster.

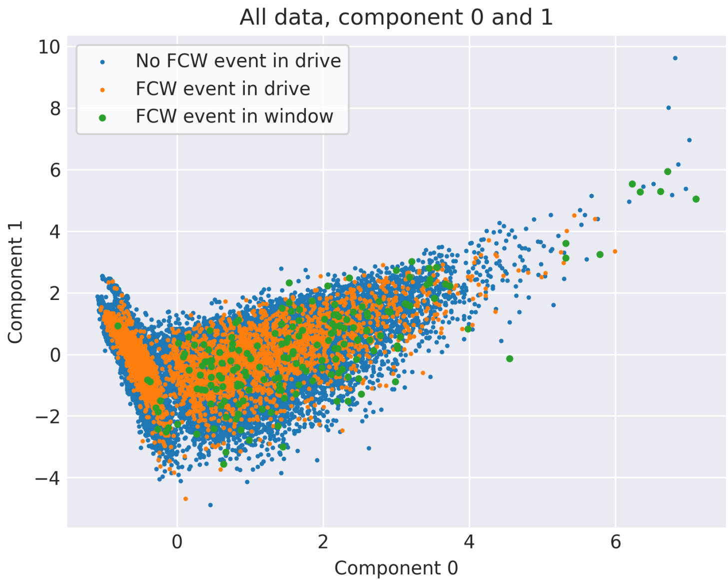


Figure 4.9: Distribution of data for component 0 and 1 when the only cleaning made is to remove NaNs and constant features. The data is colour coded with data points from the no-FCW set in blue and the data points from the FCW set in either yellow or green. The green data points correspond to the windows that contains an event, while the yellow points correspond to windows with no event although some event happens during the same drive.

The same distribution is shown in Figure 4.10, but here the importance of having a primary target becomes clear. Component 0 describes this feature very well, since all windows in the left cluster are either last in their drive and/or has no primary target. It is possible but not certain that the windows marked as outliers or last windows in the left cluster also has no target. Only a few windows in the right cluster are without target.

Figure 4.11 shows the importance of calculating all windows from time series of the same length. In this plot the same cleaning as for the previous plot was used, which does not include deleting the last window of each drive before applying PCA. Therefore, the variance caused by different length of windows resulted in a component solely describing the length of the windows. When all last windows was removed before applying PCA, none of the top components showed any distinct differences between last windows and other windows.

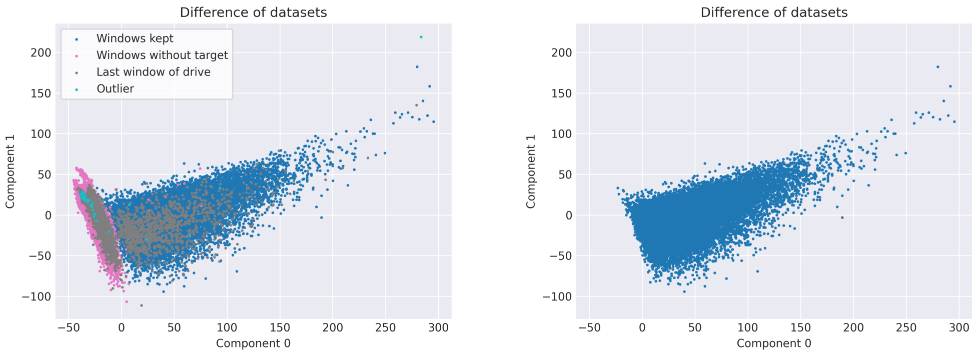


Figure 4.10: The figures show the difference between the windows used for the two different cleaning strategies, shown using the first two components from the version with less cleaning. All windows from drives with and without events are included and no distinction between them are made. The left cluster in the left plot completely disappear when deleting all windows that has no target and/or are last in their drive. The right plot shows which windows are kept with the other strategy. A file with an outlier with strange behaviour in many components are also excluded since it might be faulted.

In Figure 4.12 it is shown how the first window of each drive has a different distribution than the windows of other order. In the first plot the simple cleaning was applied before doing PCA, and component 1 clearly contains some property that varies less for the first windows. The second plot shows results from when the full cleaning described in Section 3.2.3 was applied before doing PCA. Here the main cluster of first windows instead has higher variance than other windows.

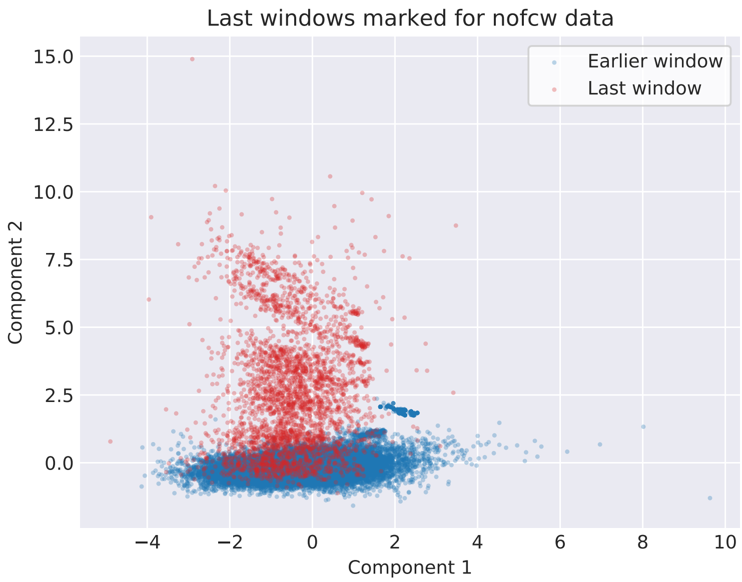
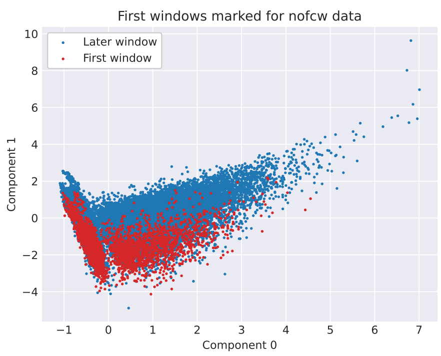


Figure 4.11: Distribution of data for component 1 and 2 before deleting the windows with no primary target. Only data from drives without events are used and the windows that are last in each drive is marked.



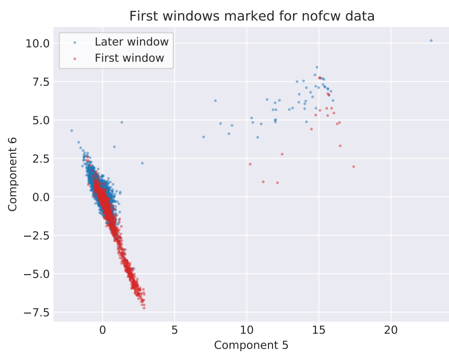


Figure 4.12: The left plot shows the distribution of data for component 0 and 1 with small amount of cleaning. Only data from drives without events are used and the windows that are last in each drive is marked. The right plot shows the same thing but for component 5 and 6 when all cleaning is applied.

5
Discussion

This chapter aims to discuss and interpret the results presented in Chapter 4. In the first section the choice of using only 60% of the total explained variance when analysing the data is discussed. Thereafter the clustering results are explored and a discussion of what could have influenced the cluster results follows. The consequences of how information is represented in the signals are covered in next section. Finally, some examples of possible future work is presented.

5.1 Information acquired from PCA

Before any conclusion can be drawn from the different principal components and the clustering analysis done in Chapter 4, one first must assess the reasoning for discarding a majority of generated components as done in this thesis. Looking at Figure 4.1 one can see that the variance in each component drops significantly for the first few components though as much as 53 components are needed to account for just 60% of the overall variance in the data. 60% might be regarded as a quite low threshold since it means that the remaining 40% of the variance is disregarded completely. Though, it is important to consider the high dimensionality of the data, combined with the fact that some features produced by tsfresh are not expected to be of interest. These “low-priority” features are thus mostly thought to be noise which, given the large amount of features, will have a significant contribution to the overall variance. Therefore, the somewhat restrictive threshold is motivated.

5.2 Takeaways from the cluster analysis

The results obtained from clustering the data points in the set with only routine driving, presented in Section 4.3, indicate that no natural groups can be found in the data set. The scores obtained for the set of drives including at least one FCW event further proves this, and applying the labels to the time series around each event gave even worse results. This indicates that the characteristics that allowed some forced clustering by k-means and agglomerative clustering in the routine data did not have anything to do with the behaviour around near-collision events.

The attempt to cluster the time series around the event gave better results with more even sizes of the two clusters and a higher Silhouette score, presented in Section 4.4.1. However, translating the clusters to the data point representation once again showed no pattern between the behaviour around critical events and the behaviour in other situations as shown in Figure 4.7 and 4.8. The clusters have approximately the same distribution in all of the top components which indicates that the location of the data points from a single drive can not tell which cluster the driver should belong to when a near-collision event occurs, since both clusters are as likely for all component values.

One explanation for this result, could be that no correlation actually exists between the behaviour at routine driving and near-collision events. This would mean that it is not possible to train a model that uses routine driving to predict behaviour at critical events such as near-collisions and collisions, since any person could behave in any way in a critical situation no matter how they behave in routine situations. Training on only critical data from collisions and near-collision events could give a better prediction of future critical events. That would however not be optimal, since critical data only makes up a very small amount of the total driving time. Due to this it would take a lot more time before the system would be able to adapt to a new driver. It would also be beneficial if the model could adapt before the first critical event occurs, but this is impossible if the model needs critical data to adapt.

5.3 Factors influencing the cluster analysis

Naturally, the lack of conclusive results regarding the connection between routine driving and FCW events in this investigation raises several questions regarding the methodology. For instance, one might argue that some other time window than 5 minutes should give a better result, or that a certain feature calculation should be excluded based on some assumption of it being irrelevant. Furthermore, the dataset itself includes several more signals not used in this project. An argument can be made that some of these signals might have been of interest to this investigation.

A motivation behind the quite liberal inclusion of features in this investigation is to be impartial to what combination of signal characteristics would be the most important. It is still quite unclear how behavior during routine driving is coupled to the behaviour in the small time frame just before a critical event or near-collision. Thus, avoiding to introduce any bias regarding this would be beneficial. The downside to including this many features is that too many features, with too little information of interest, makes the data analysis more difficult. There is a higher risk of several features being unimportant or linearly dependent of each other which is undesired. In essence, the selected features are not expected to yield a good model as is, but rather to help find out which of those features that might.

As for the choice of the time window of 5 minutes, it is hard to say if some other time window would yield a significantly different result. An inherent aspect of analyzing time series by the means of using aggregated values in this way is that for many features, the sense of order between signal values is lost. For a simple feature such as the mean of a time series for example, the concept of time is completely disregarded. This is not true for every feature calculated over the signals, but it is a further source of the problem discussed above, about the relevance of some features being difficult to assess. There is also a significant disconnection between two consecutive time windows, which in some cases might pose a problem. While being neutral to in which order certain events occur can benefit the comparison, there might also be a loss of information between windows that could have been useful. One must in this case consider the characteristic time for the phenomena of interest. As described in Section 1.3, driving behaviour can be categorized into different levels. By choosing a time window of 5 minutes, there is risk of losing information about phenomena that span over longer times. The events with a shorter characteristic time on the other hand, will become harder to detect in a longer window. This would be a further source of noise which hinders the data analysis.

Another factor that was found which hindered the analysis, was tied to the choice of using all time series data from start to finish, without first identifying the various traffic situations encountered. After investigating the first few principal components described in Table 4.1, it became clear that for these components, small variations in behaviour were seemingly overshadowed by the variability of different traffic situations. Thus it became unfeasible to compare different drivers in this regard as they might, and most likely will, encounter completely different traffic scenarios. It cannot be ruled out that the variation seen in the transformed data is the result of the variability in traffic situation, and how any reasonable driver would react in such scenario. Thus, difference in behavior is very likely to have trivial explanations such as that one driver encountered more traffic than some other, or that there was a difference in speed limit. The possible actions to prevent these issues are discussed further in Future work, Section 5.5.

Another possible explanation for the result could be the non-linearity of the data. As PCA finds linearly correlated features, it does not help very well in trying to find more complicated non-linear correlations. As discussed in Section 1.3, the timing of brake reactions is not just dependent on some inherent reaction time of the driver. The timing is also heavily dependent on other factors, some of which might be non-linear combinations of the investigated features, but perhaps also features not included in this study. Thus, these circumstances may hinder the clustering on the data acquired from the PCA.

5.4 Representation of information

In signals that are continuous but have a default value as well, the default value is often very different than the normal values. This is to avoid confusion, which is a good thing in general, but it becomes a problem for the method in this investigation. In the signals this means that the value changes drastically when switching between default and not default, and the difference for different non-default values becomes very tiny in comparison. When handling these signals in the same way as fully continuous signals, this makes it easy to notice in the data when a target appears and disappears, but very hard to notice the fine difference between close and way too close.

Cleaning the data before applying tsfresh by making the signals more similar and comparable, and different states in each signal fairer in how they are represented, could change the focus from the presentation to the actual information. This could change which features becomes most important for the top components when we run PCA on the cleaned signals. It is possible that this would help with dividing the data more and reduce the amount of noise that is now included even among the top components.

5.5 Future work

For future work, there are several options that can be investigated further and in this section some alternatives are suggested which might be suitable.

A considerable problem discussed above is that the current data representation, and the way the data is transformed, limits the ability to account for different traffic situations. A way to circumvent this, could be to first try classifying predefined traffic situations during routine driving, which could then be used to compare drivers encountering the same scenario. This is reminiscent of trying to keep everything constant except the variable under investigation; the variable being the driver in the car and the traffic situation the thing to be constant. The question that remains is how to classify these simpler situations and to know which situations carry the most information about a driver's behaviour during FCW events. A reasonable hypothesis could be that the situations that most closely resembles those of FCW events while still being regarded as routine driving could be useful.

Another option for future work, would be to use some other method of analysis and not rely on feature extraction using tsfresh. This approach, however, is accompanied with several new problems that must be solved. A suggestion could be to use something like a Long-Short Term Memory (LSTM) network or some other recurrent neural network for classification or regression of time series. LSTM networks have been used in the past to classify driving data [23]–[25], though in this case it is somewhat unclear what is desired in terms of which variable or quantity to classify or fit. Looking at the problem from a machine learning perspective, the end-goal for an FCW system is to classify whether a warning should be sent or not. The predicament lies in that ground-truth data is difficult to acquire. This data is needed for training such a system and it is difficult to determine a driver's attitude towards an issued warning without surveying. If one could better determine the rate of collision warnings regarded as false positives or false negatives, that is, being regarded as too early or too late, one could perhaps use this information when training and verifying a proposed adaptive model in the future.

A third option could be to only use FCW events when evaluating the adaptation. This is somewhat similar to the approach of finding a better tool for determining true and false positives, as we are only looking at the short timeframe in the vicinity of the FCW warning. Though, not making use of the routine driving behaviour is accompanied with the issue discussed previously, about not being able to adapt before an actual event has happened. It is also unclear how many FCW events would have to be encountered before an adaptation can be made. This poses another problem, as FCW events occur rather infrequent. However, their value in terms of useful information is very high in contrast to routine driving. Therefore, a suggestion would be to use a combination of routine driving data while also giving previous FCW events considerable attention when adapting FCW timings in the future.

6
Conclusion

Given the signals and methods used in this investigation, no correlation could be found between routine and critical driving behaviour. None of the clustering methods used found any clusters of relevance and the comparison of clustering scores for the windows and the event time series showed that the forced patterns found in one of the datasets did not translate well to the other.

The aim was to investigate if a correlation exists between routine and critical driving behaviour, such that an adaptive model can be created that uses routine driving to predict how the driver would react in a critical situation. With no such correlation between driving states, it is not possible to create such an adaptive model. A further investigation is needed to determine if changing the methodology or including more or other information in the data set could give better results.

The investigation showed that the data set is very dependent on the situation, and sensible to the different representations of information that different signals use. A further knowledge of the traffic situation the car is currently in could help in making the comparison more fair, and might make the behaviour of the driver more prominent. Furthermore, cleaning the time series data more before using tsfresh could be beneficial in preventing the signal representation from taking focus from the actual information.

Bibliography

[1] World Health Organization and issuing body, *Global status report on road safety 2018*. 2018, OCLC: 1247150642, ISBN: 978-92-4-069887-1.

[2] *Safety Vision | Volvo Car Group*. [Online]. Available: https://group.volvocars.com:443/company/safety-vision (visited on 2021-05-11).

[3] R. Kiefer, D. LeBlanc, M. Palmer, J. Salinger, R. Deering, and M. Shulman, "DEVELOPMENT AND VALIDATION OF FUNCTIONAL DEFINITIONS AND EVALUATION PROCEDURES FOR COLLISION WARNING/AVOIDANCE SYSTEMS," 1999-08, Number: HS-808 964, [Online]. Available: https://trid.trb.org/view/724278 (visited on 2021-01-27).

[4] A. H. Jamson, F. C. H. Lai, and O. M. J. Carsten, "Potential benefits of an adaptive forward collision warning system," *Transportation Research Part C: Emerging Technologies*, vol. 16, no. 4, pp. 471–484, 2008-08, ISSN: 0968-090X. DOI: 10.1016/j.trc.2007.09.003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X07000721 (visited on 2021-04-16).

[5] Volvo Car Corporation, "XC60 Recharge Owner's Manual," Tech. Rep., 2021. [Online]. Available: https://volvornt.harte-hanks.com/manuals/2021/XC60_Recharge_OwnersManual_MY21_en-US_TP32117.pdf (visited on 2021-05-28).

[6] J. J. Gibson and L. E. Crooks, "A Theoretical Field-Analysis of Automobile-Driving," *The American Journal of Psychology*, vol. 51, no. 3, pp. 453–471, 1938, Publisher: University of Illinois Press, ISSN: 0002-9556. DOI: 10.2307/1416145. [Online]. Available: https://www.jstor.org/stable/1416145 (visited on 2021-04-22).

[7] M. Plöchl and J. Edelmann, "Driver models in automobile dynamics application," *Vehicle System Dynamics*, vol. 45, no. 7-8, pp. 699–741, 2007-07, ISSN: 0042-3114, 1744-5159. DOI: 10.1080/00423110701432482. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/00423110701432482 (visited on 2021-05-21).

[8] G. M. Markkula, O. Benderius, K. Wolff, and M. Wahde, "A Review of Near-Collision Driver Behavior Models," *Human Factors*, vol. 54, no. 6, pp. 1117–1143, 2012, ISSN: 0018-7208. DOI: 10.1177/0018720812448474. [Online]. Available: https://research.chalmers.se/en/publication/171559 (visited on 2021-01-26).

[9] J. A. Michon, "A Critical View of Driver Behavior Models: What Do We Know, What Should We Do?" In *Human Behavior and Traffic Safety*, L. Evans and

R. C. Schwing, Eds., Boston, MA: Springer US, 1985, pp. 485–524, ISBN: 978-1-4613-2173-6. DOI: 10.1007/978-1-4613-2173-6_19. [Online]. Available: https://doi.org/10.1007/978-1-4613-2173-6_19 (visited on 2021-04-19).

[10] G. Markkula, “Modeling driver control behavior in both routine and near-accident driving,” *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 58, no. 1, pp. 879–883, 2014-09, Publisher: SAGE Publications Inc, ISSN: 2169-5067. DOI: 10.1177/1541931214581185. [Online]. Available: https://doi.org/10.1177/1541931214581185 (visited on 2021-05-12).

[11] G. Markkula, O. Benderius, and M. Wahde, “Comparing and validating models of driver steering behaviour in collision avoidance and vehicle stabilisation,” *Vehicle System Dynamics*, vol. 52, no. 12, pp. 1658–1680, 2014-12, ISSN: 0042-3114, 1744-5159. DOI: 10.1080/00423114.2014.954589. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/00423114.2014.954589 (visited on 2021-05-30).

[12] G. Markkula, J. Engström, J. Lodin, J. Bärgman, and T. Victor, “A farewell to brake reaction times? Kinematics-dependent brake response in naturalistic rear-end emergencies,” *Accident Analysis & Prevention*, vol. 95, pp. 209–226, 2016-10, ISSN: 0001-4575. DOI: 10.1016/j.aap.2016.07.007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0001457516302366 (visited on 2021-01-26).

[13] M. Ljung Aust, J. Engström, and M. Viström, “Effects of forward collision warning and repeated event exposure on emergency braking,” *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 18, pp. 34–46, 2013-05, ISSN: 13698478. DOI: 10.1016/j.trf.2012.12.010. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1369847813000065 (visited on 2021-01-26).

[14] D. C. Lay, S. R. Lay, and J. McDonald, *Linear algebra and its applications.* 2018, OCLC: 1050756233, ISBN: 978-1-292-09223-2.

[15] *2.3. Clustering scikit-learn 0.24.2 documentation.* [Online]. Available: https://scikit-learn.org/stable/modules/clustering (visited on 2021-05-24).

[16] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, and E. Keogh, “Experimental comparison of representation methods and distance measures for time series data,” *Data Mining and Knowledge Discovery*, vol. 26, no. 2, pp. 275–309, 2013-03, ISSN: 1384-5810, 1573-756X. DOI: 10.1007/s10618-012-0250-5. [Online]. Available: http://link.springer.com/10.1007/s10618-012-0250-5 (visited on 2021-05-24).

[17] P. Senin, “Dynamic time warping algorithm review,” *Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA*, vol. 855, no. 1-23, p. 40, 2008.

[18] R. Bellman and R. Kalaba, “On adaptive control processes,” *IRE Transactions on Automatic Control*, vol. 4, no. 2, pp. 1–9, 1959-11, ISSN: 0096-199X. DOI: 10.1109/TAC.1959.1104847. [Online]. Available: http://ieeexplore.ieee.org/document/1104847/ (visited on 2021-05-24).

[19] M. Christ, N. Braun, J. Neuffer, and A. W. Kempa-Liehr, “Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests (tsfresh A Python

package)," *Neurocomputing*, vol. 307, pp. 72–77, 2018-09, ISSN: 09252312. DOI: 10.1016/j.neucom.2018.03.067. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0925231218304843 (visited on 2021-05-19).

[20] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[21] *Sklearn.cluster.kmeans*. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans (visited on 2021-06-20).

[22] *Sklearn.cluster.agglomerativeclustering*. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering (visited on 2021-06-20).

[23] J. S. Wijnands, J. Thompson, G. D. P. A. Aschwanden, and M. Stevenson, "Identifying behavioural change among drivers using Long Short-Term Memory recurrent neural networks," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 53, pp. 34–49, 2018-02, ISSN: 1369-8478. DOI: 10.1016/j.trf.2017.12.006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1369847817301183 (visited on 2021-02-16).

[24] K. Saleh, M. Hossny, and S. Nahavandi, "Driving behavior classification based on sensor data fusion using LSTM recurrent neural networks," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, ISSN: 2153-0017, 2017-10, pp. 1–6. DOI: 10.1109/ITSC.2017.8317835.

[25] F. Rundo, "Deep LSTM with Dynamic Time Warping Processing Framework: A Novel Advanced Algorithm with Biosensor System for an Efficient Car-Driver Recognition," *Electronics*, vol. 9, no. 4, p. 616, 2020-04, Number: 4 Publisher: Multidisciplinary Digital Publishing Institute. DOI: 10.3390/electronics9040616. [Online]. Available: https://www.mdpi.com/2079-9292/9/4/616 (visited on 2021-02-17).

DEPARTMENT OF MATHEMATICAL SCIENCES
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2021
www.chalmers.se