

## Overtake Detection in Trucks Using CAN Bus Signals: A Comparative Study of Machine Learning Methods

Fernando Alonso-Fernandez (feralo@hh.se), Talha Hanif Butt  
(talha-hanif.butt@hh.se), Prayag Tiwari (prayag.tiwari@hh.se)

School of Information Science, Computer and Electrical Engineering, Halmstad University, Box 823, Halmstad SE 301-18, Sweden

**Corresponding author:**

Fernando Alonso-Fernandez

School of Information Science, Computer and Electrical Engineering, Halmstad University, Box 823, Halmstad SE 301-18, Sweden

Tel: +46 729 77 35 29

Email: feralo@hh.se

# Overtake Detection in Trucks Using CAN Bus Signals: A Comparative Study of Machine Learning Methods

Fernando Alonso-Fernandez\*, Talha Hanif Butt, Prayag Tiwari

*School of Information Science, Computer and Electrical Engineering, Halmstad University,  
Box 823, Halmstad SE 301-18, Sweden*

---

## Abstract

Safe overtaking manoeuvres in trucks are vital for preventing accidents and ensuring efficient traffic flow. Accurate prediction of such manoeuvres is essential for Advanced Driver Assistance Systems (ADAS) to make timely and informed decisions. In this study, we focus on overtaking detection using Controller Area Network (CAN) bus data collected from five in-service trucks provided by the Volvo Group. We evaluate three common classifiers for vehicle manoeuvre detection, Artificial Neural Networks (ANN), Random Forest (RF), and Support Vector Machines (SVM), and analyse how different preprocessing configurations affect performance. We find that variability in traffic conditions strongly influences the signal patterns, particularly in the no-overtake class, affecting classification performance if training data lacks adequate diversity. Since the data were collected under unconstrained, real-world conditions, class diversity cannot be guaranteed a priori. However, training with data from multiple vehicles improves generalisation and reduces condition-specific bias. Our per-truck analysis also reveals that classification accuracy, especially for overtakes, depends on the amount of training data per vehicle. To address this, we apply a score-level fusion strategy, which yields the best per-truck performance across most cases. Overall, we achieve an accuracy via fusion of  $TNR=93\%$  (True Negative Rate) and  $TPR=86.5\%$  (True Positive Rate). This research has been

---

\*Corresponding author.

*Email addresses:* `feralo@hh.se` (Fernando Alonso-Fernandez), `talha-hanif.butt@hh.se` (Talha Hanif Butt), `prayag.tiwari@hh.se` (Prayag Tiwari)

part of the BIG FUN project, which explores how Artificial Intelligence can be applied to logged vehicle data to understand and predict driver behaviour, particularly in relation to Camera Monitor Systems (CMS), being introduced as digital replacements for traditional exterior mirrors.

*Keywords:* Machine Learning, Overtake Prediction, CAN Bus Data, Truck Maneuvers, Advanced Driver Assistance Systems

---

## 1. Introduction

Developing Advanced Driver Assistance Systems (ADAS) is a major focus in artificial intelligence research for the automotive industry. These systems use data from various sensors, such as cameras, LiDAR, radar, biosensors, and vehicle networks, to alert drivers of potential hazards or autonomously adjust vehicle behaviour to prevent accidents (Badue et al., 2021; Kumar et al., 2020). A key requirement for effective ADAS is the ability to analyse environmental and vehicular data, including speed, acceleration, and lane position, for both the ego and surrounding vehicles (Hasan et al., 2021). Additionally, many systems must assess the driver's intention in order to issue timely warnings or override inputs if needed (Xing et al., 2019; Zhang et al., 2021). Understanding driver behaviour in contexts such as overtaking, braking, turning, or lane changes is thus essential to improving ADAS performance and vehicle safety (Mozaffari et al., 2020). Such insights support systems responsible for lane departure warning, blind-spot detection, collision avoidance, and adaptive cruise control, among others.

In this work, we focus on the design and evaluation of an automatic overtaking detection system for trucks. Compared to other vehicles, trucks pose greater risks due to their size and mass, making truck-related accidents potentially more severe. These accidents can lead to more fatalities, traffic congestion, and significant economic loss due to cargo delays (Fu et al., 2021; McKnight & Bahouth, 2020). To address this and contribute to improved safety and efficiency in freight transportation, we aim to enhance the ability to detect and monitor overtaking behaviour in trucks using CAN bus signals. These

signals are available onboard without requiring additional hardware, such as cameras or biosensors (Garcia et al., 2020; Neumeier et al., 2019), making them a cost-effective and privacy-friendly option for deployment. This avoids concerns related to video-based monitoring and supports scalable ADAS solutions for heavy-duty vehicles.

For this study, we utilise CAN data collected from actual, in-service trucks, generously provided by the Volvo Group, a key research partner in this work. This research has been conducted within the framework of the BIG FUN project (Big Data-Powered End User Function Development), which explores how Artificial Intelligence (AI) can be applied to logged vehicle data to identify and predict driver behaviour, particularly in relation to Camera Monitor Systems (CMS), which are being introduced as replacements for traditional exterior mirrors. While CMS technologies offer clear advantages in terms of visibility and aerodynamics, they may also influence driver behaviour in ways that are not yet fully understood. In addition, they provide new opportunities for analysing logged data, as these systems inherently generate digital video, which can be combined with other sensor data streams. Overtaking manoeuvres are especially relevant in this context, as indirect vision, whether provided by CMS or traditional mirrors, plays a critical role in driver decision-making during such actions. Motivated by this, our study focuses on the detection of overtaking events using CAN bus data, which is available both in new CMS trucks and in older vehicles with conventional mirrors. Our long-term aim is to enable behavioural analysis and comparison between the two configurations, as drivers adapt to CMS and begin to rely on digital video instead of traditional optical systems.

Our overtaking detection approach using CAN data, built on popular machine learning techniques, extends methods from conclusions and recommendations from related literature on vehicle manoeuvre detection (Xing et al., 2019). Although we use standard classifiers like ANN, RF, and SVM, our primary contribution lies in applying them in what is, to the best of our knowledge, one of the first studies on overtaking detection in trucks using real CAN bus data. While

most previous works rely on a single classifier, we systematically compare these well-established, yet diverse models under the same experimental conditions and analyse how their performance is influenced by different preprocessing configurations. This also allows to explore the benefits of classifier fusion, which, as our results show, contributes to improved robustness across different trucks. Our work also encompasses collecting and annotating a dataset to enable training and evaluation of the system. Most prior works deal with other manoeuvres (e.g., lane changes) or use simulated or passenger vehicle data, which may not generalise to the context of commercial trucks. Here, we present an original dataset of annotated real-world CAN signals from five operational trucks in naturalistic driving scenarios. As illustrated in Figure 1, several CAN signals, which measure various aspects of vehicle operation (such as pedal positions, speed, etc.), are continuously logged. When a predefined trigger condition is met (indicated by an arrow in the figure), a crop of these signals is extracted and passed to a classifier, which determines whether the crop corresponds to an overtake. The signals in each crop are usually pre-processed using a sliding window approach, where metrics such as the average and standard deviation of the signals are computed within the window.

A preliminary version of this work was presented at a conference (Butt et al., 2024), where a smaller dataset (712 files from 3 trucks) was employed. In this paper, we expand the dataset to 1247 files, increasing the number of trucks analysed from 3 to 5. Additionally, the previous work employed a fixed configuration for the crop and sliding window sizes. Here, we conduct an in-depth investigation on how different preprocessing and crop configurations affect detection. In particular, we analyse the influence of the crop size, sliding window size, and metrics computed within each window. This analysis is motivated by the absence of studies dedicated to overtaking detection in trucks and the lack of consensus on optimal configurations of these parameters in related research, as discussed in (Xing et al., 2019; Khairdoost et al., 2020), and observed by the different parameter choices in works such as (Kim et al., 2017; Zhang & Fu, 2020; Khairdoost et al., 2020). Some studies even consider the use of dynamic

crops to cope with manoeuvres with varying time lengths (Zheng & Hansen, 2017; Das et al., 2020, 2023). The present paper also provides insights missing in the previous version, such as the impact of the number of trucks used to train the classifiers, the visual patterns of the different CAN signals, or the per-truck performance of the system. Finally, we demonstrate that classifier fusion can enhance per-truck accuracy, improving system generalisation and robustness.

The rest of the paper is organised as follows. Section 2 describes related work. Section 3 extensively describes the proposed system, including the database, machine learning methods, and experimental protocol. The experimental results are presented and discussed in Section 4. Finally, the conclusions are given in Section 5.

Table 1: Summary of related works on overtake, lane change and turning detection using CAN signals. OV=Overtake. LC=Lane Change. LK=Lane Keep. TU=Turning. GS=Going Straight. Other acronyms are defined in the text.

Task	Work	Vehicle	CAN signals	Preprocessing of CAN signals	Classifier	Data	Results (CAN only)
(Blaschke et al., 2008)	Car	5	Raw signals	Fuzzy Logic	43 OV, 55 no OV 156 OV, 16 no OV		TPR=95%, TNR=93% TPR=94.2%, TNR=100%
OV (Stefansson et al., 2020)	Simulator	2	Raw signals	Kalman filter	52 OV	n/a	n/a
(Lin et al., 2021)	Golf cart	5	Raw signals	Vision	n/a	n/a	n/a
<b>This work</b>	<b>Truck</b>	10	Rule for OV candidates, then mean, variance.	ANN, RF, SVM	382 OV, 865 no OV		TPR=86.5%, TNR=93% (SVM+RF) TPR=86.2%, TNR=95.1% (RF)
<hr/>							
(Kim et al., 2017)	Simulator	11	Mean, variance, PCA, Crops=0.5 sec, window=0.2 sec	SVM	n/a		Accuracy=90.97%
(Zheng & Hansen, 2017)	Car	4	LC candidates and TU removal via steering angle information	DTW+KNN HMM	UTDrive dataset 122/209 left/right LC, 87/404 left/right TU, 2943 LK		Accuracy=58.92%
LC (Das et al., 2020)	Car	4	LC candidates using cameras, then mean, max, min, std	RF, SVM ANN, XGBoost	SHRP2 dataset 1200 LC, 2400 no LC		AUC=0.81-0.84
(Guo et al., 2022)	Car	2	Sliding window prediction every 0.1 sec	Autencoder, Transformer	DASI dataset 2005/1480 left/right LC, 34850 LK		Accuracy=74.7-76.3% n/a (combined vision, GPS and CAN)
(Das et al., 2023)	Car	4	Same as (Das et al., 2020) XGboost for correlation removal	ResNet18 CNN ANN, XGBoost, CART SVM, RF, KNN, NB	SHRP2 dataset 1200 LC, 68173 no LC		CNN: Recall=82%, Accuracy=77.9% CNN: Recall=82%, Accuracy=77.9%
(Huang et al., 2019)	Car	2	Raw signals	FCN+GMM	n/a		n/a (combined vision, GPS and CAN)
TU (Zhang & Fu, 2020)	Car	4	Signals estimated from video Crops 11 sec, window=0.1 sec	Bi-LSTM	NGSIM dataset 1039/978 left/right TU, 976 GS		Accuracy=74.5, 93.5, 94.2% (3, 2, 1 sec ahead TU)
(Khairdoost et al., 2020)	Car	10	Histogram, min, avg, max Crop 2/3 sec	LSTM	RoadLAB dataset 65/40 left/right LC, 65/75 left/right TU, 80 GS		n/a (combined vision, driver gaze/head and CAN)

## 2. Related Work

The overtaking manoeuvre is one of the most critical and challenging driving actions, involving a sequence of decisions such as lane changes, acceleration and deceleration, and real-time estimation of the speed and distance of surrounding vehicles. Despite its importance, research focusing on overtaking prediction using Controller Area Network (CAN) data, especially in trucks, is scarce. The few existing studies are focused on other vehicle types, such as passenger cars or experimental prototypes (Blaschke et al., 2008; Stefansson et al., 2020; Lin et al., 2021). Because overtaking carries a high risk, no real-world dataset is available either (Dutra & Fonseca, 2022). The majority of related research concentrates on detecting other types of manoeuvres, such as turning intention at intersections or lane change prediction, a manoeuvre closely related to overtaking. Detecting these manoeuvres is done using diverse data sources (Xing et al., 2019) which measure:

- i) The vehicle dynamics, engine and driver inputs, such as speed, acceleration, pedals' position, wheel angle, etc. This is commonly captured by the Electronic Control Units (ECUs) of the vehicle.
- ii) The traffic information, such as position, distance, and motion of surrounding vehicles, or other road users like cyclists and pedestrians, as well as traffic signs and road hazards. This is typically measured by sensors such as radar, LIDAR, cameras, or ultrasonic sensors. Researchers may also employ GPS or map information, or include aspects such as temperature, weather or road condition (Kim et al., 2017).
- iii) The vehicle in relation to the surrounding environment, such as the status and warnings regarding lane departure, blind spots, collisions, etc. These measurements are usually enabled by the data sources of *i*) and *ii*).
- iv) The driver, such as gaze direction, head position, fatigue levels, drowsiness, distraction, etc. This is captured via eye trackers, cameras or biosensors that capture physiological data such as EEG or ECG. In some cases,

studies include demographic information of the driver, such as age, gender, driving experience, etc., although this information may proceed as part of annotated data collection, rather than actual automatic estimation (Das et al., 2020).

The Controller Area Network (CAN) is a communication protocol that aggregates vehicle data to enable its efficient real-time operation and coordination of its subsystems (engine, transmission, braking, etc.). It is also used for vehicle diagnostics, maintenance, and performance monitoring. It typically captures aspects related to *i-iii*) above, and even if these can engage sensors such as cameras, radar, or LIDAR, the data transmitted in the CAN protocol only contains processed outputs from these sources (e.g., distances to obstacles or their velocity) rather than the raw sensor streams. We contrast this with other research which explicitly uses such raw data streams for manoeuvre detection, e.g. (Xing et al., 2019). Since they are not readily available from CAN data, external sensors placed inside or outside the vehicle are usually engaged. In this section, we thus review relevant studies emphasising the utilisation of CAN data only. Table 1 presents a summary of the relevant works covered in this Section.

Another related field of interest is the intention recognition of surrounding vehicles in intelligent and autonomous systems. These approaches aim to assess, for example, whether it is an appropriate time for the ego vehicle to perform a manoeuvre such as changing lanes (Wang et al., 2022) or overtaking (Ortega et al., 2024) based on surrounding traffic conditions. Similarly, they attempt to predict the intentions of nearby vehicles, such as whether another vehicle will change lanes (Liu et al., 2023), cross or turn at an intersection (Pourjafari et al., 2024), or initiate an overtaking manoeuvre (Lin et al., 2020). However, we view this line of research as following a different focus. Rather than detecting manoeuvres being executed by the current vehicle, these systems infer the future actions of other road users. In addition, CAN data primarily captures internal vehicle information, so it could provide very little evidence of such phenomena. As a result, such approaches typically rely on external perception data from

cameras, LiDAR, or radar to observe the environment. Therefore, we consider those approaches on intention recognition of other vehicles to be outside the scope of this paper.

### *2.1. Overtake detection*

As mentioned earlier, the literature on overtaking detection is limited. Most studies discussed in this section focus on assessing whether overtaking is safe. We include them primarily because they utilise CAN data to analyse the manoeuvre.

(Blaschke et al., 2008) presented one of the first studies focused on predicting overtaking manoeuvres using CAN bus data. In a first stage, they involved 28 drivers between 22-65 years old on country roads. A vehicle ahead drove partly at the same speed and partly 30 km/h slower than the recommended speed limit. The data generated included 43 overtaking and 55 no overtaking manoeuvres, achieving an accuracy in detecting these two manoeuvres of TPR=95% and TNR=93%, respectively. Their fuzzy logic approach integrated 5 indicators to predict overtaking behaviour (distance and speed difference to the preceding vehicle, brake pressure, accelerator pedal pressure and accelerator pedal speed). In addition, the algorithm is applied only to time points with a speed above 60 km/h. In a second stage, they obtained 156 overtake and 16 no overtaking manoeuvres on country roads, motorways and city traffic, achieving TPR=94.2% and TNR=100%. An observed limitation compared to our present study is the relatively small dataset.

(Stefansson et al., 2020) proposed models for the decision-making process of human drivers. The authors introduced a mathematical formulation of the overtaking problem and then proposed decision models judging whether a vehicle should overtake or not based on inputs such as a slow-moving vehicle ahead or an approaching vehicle in the other direction on a two-way road. The decision models were based on a Kalman filter estimator with distance and velocity difference between vehicles as input signals, and two decision rules were derived from different risk-awareness levels. An experimental testbed was also intro-

duced using a driving simulator with two drivers carrying out 52 overtaking decisions, classifying the manoeuvre as safe or unsafe.

(Lin et al., 2021) proposed a Model Predictive Control (MPC) scheme integrated with time-to-lane-crossing (TLC) estimation for autonomous vehicles (an electric prototype golf cart). Thus, the system is not really a detector of the driver’s intention, but a detector of when an autonomous vehicle can safely overtake others. They used 5 CAN bus signals for communication between sensors, actuators, and the vehicle control unit (VCU), but the system relied heavily on a vision-based system to detect lane lines and other vehicles to estimate TLC and decide whether to overtake or stay in lane. This limits its applicability to pure CAN data environments. The system determined the optimal throttle, brake, and steering commands to ensure safe longitudinal and lateral vehicle control.

In the present paper, we used 10 onboard CAN signals to predict overtakes with three machine learning methods, ANN, RF and SVM. While these classifiers are popular choices in vehicle manoeuvre detection, appearing in several studies of Table 1, our study is, to the best of our knowledge, the first to focus specifically on trucks using real-world CAN data from operational heavy-duty vehicles. Unlike most previous studies that rely on a single classifier, we compare multiple models and also evaluate their fusion. Data were collected from five trucks, comprising 382 overtake and 865 no-overtake segments. A precondition rule was used to identify potential overtakes, which were then manually annotated via video review to enable classifier training and evaluation. After such data labelling, video data was not further considered, so the developed predictors could rely only on CAN data. Unlike previous works that used fixed crops and sliding window sizes, and given the lack of consensus regarding their optimal values (Xing et al., 2019; Khairstoost et al., 2020), here we systematically explored multiple configurations to assess their impact on performance, and found that the optimal configuration varies depending on the classifier used. Our best results overall were obtained with RF (TPR=86.2%, TNR=95.1%), while fusion with SVM improved robustness across vehicles, achieving TPR=86.5% and

TNR=93%.

## 2.2. Lane change detection

The majority of related studies are in the context of lane change prediction. The task is usually modelled using left lane change, right lane change, and lane keeping as classification classes. Detecting such manoeuvres can prevent accidents in scenarios with limited visibility, heavy traffic, or when merging, enabling timely alerts to avoid potential collisions. (Xing et al., 2019) carried out a state-of-art literature review, focusing on highways. Being from 2019, the studies reported mostly correspond to the pre-deep-learning era, with algorithms covering a wide range of methods, including Support Vector Machines (SVM), Hidden Markov Models (HMM), Artificial Neural Networks (ANN) and others, and just one study applying Long Short-Term Memory (LSTM) networks. The obtained accuracies vary, with some studies surpassing 90% and reaching up to 98-100%. The prediction horizon was, at maximum, 3.5 seconds before the lane change, with many studies employing 1 second or less.

Among the studies reported in (Xing et al., 2019), very few used only vehicle information from CAN signals as data input, with many relying on other dedicated sensors to capture the data aspects *i-iv*) mentioned at the beginning of this section. Some do not even use CAN data at all. Regarding databases, some works employed naturalist on-road data, while others employed a driving simulator. However, the data is unavailable in general, making replication difficult. In what follows, we report more recent studies on lane change prediction that are not included in the mentioned literature review paper.

(Kim et al., 2017) used different combinations of 11 onboard signals to detect a lane change with an SVM classifier and data captured with a PC driving simulator. The signals were processed with a time crop and an overlapping sliding window of 0.5 and 0.2 seconds, respectively. As features, they used the average, variance and principal components (PCA) of the windowed signals. The simulation contained different road conditions (dry, gravel, wet and snowy), also estimated from the signals with an ANN. Accuracies in lane change detection

varied between 90 and 97% with the best-observed combination of signals.

(Zheng & Hansen, 2017) applied two machine learning methods, one based on Dynamic Time Warping (DTW) features with a k-Nearest Neighbor (k-NN) classifier and another using an HMM classifier. They considered four CAN signals, including vehicle speed, steering angle, brake pressure, and engine rpm. Data came from a subset of the UTDrive naturalistic driving corpus, comprising 14 females and 44 males driving a Toyota RAV4 vehicle, with a mix of local and residential roads having 2-3 lanes and an average speed of 40 mph. In the first stage, left/right turn manoeuvres are filtered out by seeking very large steering angle changes. Then, potential boundaries between lane-keeping (straight wheels) and lane-changing (non-straight wheels) are sought by applying a filter bank to the spectrogram of the steering angle signal. This is because a high variation of energy across bands has been observed to be indicative of the beginning of a lane change. Afterwards, candidate lane change segments are classified by the k-NN and HMM methods. The driving sessions of the database accounted for a wide variability of speeds, so a lane change manoeuvre may be executed over varying time lengths. The proposed spectral segmentation allowed accounting for such variability, which makes the use of fixed window segments undesirable. In addition, DTW and HMM are also inherently capable of coping with data sequences of different lengths. The obtained accuracies ranged between 58 and 92%, depending on the class and the algorithm.

(Das et al., 2020) also employed non-uniform time crops of the signals. To do so, they segmented 1200 lane-change segments from the SHRP2 Naturalistic Driving Study (NDS) dataset with a vision technique that analyses the offset of the vehicle with respect to the lane centre since the database not only contains CAN data. 2400 no lane-change segments were also extracted as negative class. The database contains a variety of weather conditions (clear, snow, rain, and fog). Lane change classification with four CAN signals was done by comparing RF, SVM, ANN, and Extreme Gradient Boosting (XGBoost) algorithms. The dataset included four vehicle signals: speed, longitudinal acceleration, lateral acceleration, and yaw rate. To account for variability in the manoeuvre dura-

tion, the mean, maximum, minimum, and standard deviation of the signals in the segments were used as features. With only CAN signals, the results obtained were AUC=0.864 (RF), 0.855 (SVM), 0.81 (ANN), and 0.859 (XGBoost)

Deep learning methods have increasingly been applied in recent years for lane change detection, although the literature employing CAN signals is very scarce (Guo et al., 2022; Das et al., 2023), being more popular the use of video data (Scheel et al., 2019; Xie et al., 2019; Nalcakan & Bastanlar, 2023) including drones (Xu et al., 2021; Chen et al., 2021), or video combined with GPS information (Xing et al., 2020).

(Guo et al., 2022) applied an autoencoder-based lane change detector and a transformer-based lane change predictor. The employed DAS1 database, which is public, included 7960 trips recorded by 98 sedans, with data coming from a forward-looking camera integrated into an after-market ADAS system (Mobileye), a GPS unit, and CAN signals. However, among the 17 signals available, only two are CAN signals (yaw rate and longitudinal acceleration), and the accuracy with only those or their impact was not explicitly assessed.

The authors of (Das et al., 2020) extended the study using a ResNet-18 Convolutional Neural Network (CNN) (Das et al., 2023). They first balanced the classes using synthetic minority oversampling (SMOTE) and random majority undersampling (RMUS), and then applied XGBoost to remove highly correlated features. Then, they converted the time sequence signals to an image, which was given to the CNN. With only CAN signals, the reported recall was 82% and the overall accuracy of the two classes was 77.9%. The CNN method was also observed to outperform other ML models in terms of recall, although precision was lower in some cases.

### *2.3. Turning detection*

Turn intention prediction is another field of research that is related. The task is usually addressed using left turn, right turn and going straight as classes. Predicting such intentions is crucial in intersections, ramps, parking areas, and roundabouts, where reduced visibility and potential collisions with other road

users pose significant risks.

(Huang et al., 2019) used a combination of multiple signals from CAN data, front and side cameras, GPS and an Inertial Measurement Unit (IMU) placed on a Toyota Lexus sedan driven for 30 hours during 14 trips in different weather and lighting conditions. The CAN signals considered included steering wheel angle and gas pedal value, which allows the acceleration or turning rate to be modelled. Different expert predictors were integrated, including a VGG CNN for image data and fully connected networks (FCN) for the remaining data. The output of the experts is concatenated and sent to another fully connected network, which outputs Gaussian Mixture Models (GMM) parameters. The paper studied the complementarity of the expert predictors, and although it did not evaluate the use of CAN data separately, dropping CAN signals was seen to produce an increase in the prediction error of 13.29%.

(Zhang & Fu, 2020) developed a hybrid model that first applied autoregressive integrated moving average (ARIMA) to estimate the lateral position, longitudinal position, speed, and acceleration of the vehicle from video data, followed by a bidirectional LSTM (Bi-LSTM) network to predict turning behaviour using the estimated signals. They used the open-source NGSIM dataset, which contains both video data and the mentioned vehicle signals as a reference, although the latter was estimated by the authors of the database from the video stream as well. They employed a total of 2993 samples of data containing left/right turns and going straight segments, obtained by analysing the heading angle of the vehicle, and then extracted a crop of 11 seconds of the entire turning process. The method is evaluated with a sliding window approach having a length of 1.5 seconds and a moving step of 0.1 seconds. The obtained average recognition rate was 74.5%, 93.5% and 94.2% at 3, 2 and 1 seconds, respectively, before initiating the manoeuvre.

(Khairdoost et al., 2020) applied LSTM to a combination of driver gaze, head position, a forward vision system on the rooftop, and CAN data from 325 segments of the RoadLAB database, captured using an equipped sedan driven by 16 drivers on a 28.5 km pre-determined city course that included urban and

suburban areas. The study considered feature aggregation across 20 frames of data captured at 30 Hz (2/3 of a second). For each crop, CAN data consisted of the histogram of steering wheel angles, the binary value of left and right turn signals, and the minimum, average and maximum vehicle speed, brake pedal pressure, gas pedal pressure and speeds of the four wheels. The precision/recall was 85.6% and 84.1%, but with the entire set of signals, not providing results with CAN data only. The study also considered the use of left/right lane change as possible additional manoeuvres, with a precision/recall over the extended set of classes of 84.2% and 82.9%.

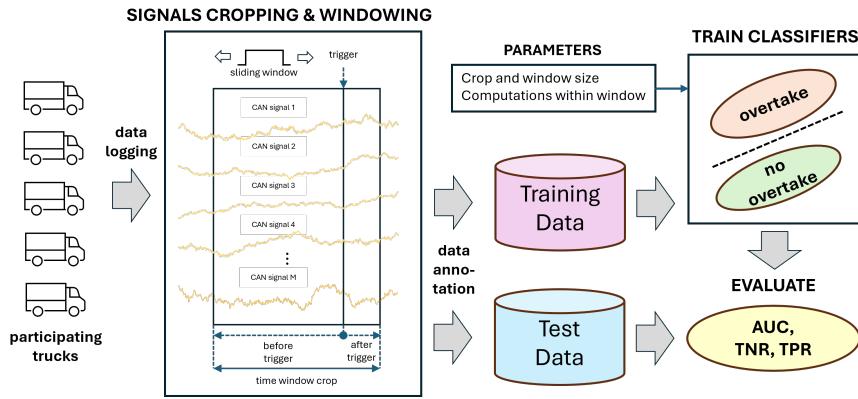


Figure 1: Overview of the overtake classification system.

### 3. Materials and Methods

This section describes the composition of our dataset, the machine learning methods and the experimental protocol employed. An overview of our overtaking classification system is shown in Figure 1. We follow standard practices commonly used in manoeuvre detection studies, including data selection based on trigger conditions, manual annotation, signal preprocessing, model training, and performance evaluation using standard metrics such as Area Under the Curve (AUC), TNR and TPR. The manual annotation of overtaking events was conducted through synchronised video review to ensure reliable ground truth labels. To validate our approach, we use a standard hold-out evaluation, where

classifiers are trained on a predefined training set until convergence and then tested on a separate, disjoint test set, ensuring that the evaluation reflects generalisation to unseen data.

### *3.1. Database*

Our database consists of data from 5 actual operating trucks normally driving around Europe, provided by Volvo Group, participating in this research. Three of the trucks are equipped with a regular physical mirror, and the two others with a mirrorless CMS (Camera Monitor System). The trucks are equipped with a data logger that captures CAN signals at 10 Hz. The common signals available in all trucks and employed for this work are listed below:

1. Position of the accelerator pedal
2. Distance to the vehicle ahead
3. Speed of the vehicle ahead
4. Relative speed difference between the vehicle and the left wheel
5. Vehicle speed
6. Vehicle lateral acceleration
7. Vehicle longitudinal acceleration
8. Lane change status of the vehicle
9. Status of the left turn indicator
10. Status of the right turn indicator

Signals 1 to 7 vary continuously within specific ranges, whereas signals 8 to 10 have discrete statuses related to activation/no activation. Other CAN signals are also available in the data logger, but they are not considered in this work since we consider that they are not informative of an overtake manoeuvre.

Table 2: Files available per truck and per class. tx denotes the truck number (given by x). class0=no-overtake. class1=overtake.

Type	Truck	class0	class1	total
<i>Mirror</i>	t1	125	417	542
	t2	163	83	246
	t3	8	11	19
	total	296	511	807
<i>CMS</i>	t4	81	342	423
	t5	5	12	17
	total	86	354	440
Both	total	382	865	1247

These include, for example, the position of the brake pedal, the position of the gear lever, or the status and speed set in the cruise control.

To avoid storage issues and to obtain segments where a potential overtaking occurs, we defined a precondition trigger so that the data logger only records when it is met. Such trigger is activated based on specific thresholds to the following signals: signal 8 (active), signal 5 (more than 50 km/h), signal 2 (less than 200 m), and signal 4 (more than 0.1 km/h). If the trigger is activated, the logger saves the CAN signals from 20 seconds before the trigger up to 45 seconds thereafter. The logger also saves video from a camera placed on the dashboard looking ahead of the vehicle. The files are later labelled manually by watching the videos and determining if the recorded segment is an overtaking or not. Afterwards, the video data is discarded for further use.

After this process, we obtained the files indicated in Table 2. Compared to our previous work (Butt et al., 2024), we have increased the number of files of trucks t1-t3 from 712 to 807. Also, we incorporate data from trucks t4-t5, which were not available in the previous paper. The precondition trigger is designed to detect when the vehicle is to change lane (signal 8), to be sufficiently

close to the vehicle ahead (signal 2), and to move laterally to the left (signal 4), which are indicative signs of a potential overtake. However, around 31% of the obtained files correspond to other driving situations where the truck can show similar dynamics. After watching the videos, it occurs, for example, when turning left at an intersection, changing lanes to leave an exit-only lane, giving way to a vehicle merging into the road, or surpassing a stopped vehicle or construction workers. The left turn indicator (signal 9) could be another possible precondition indicator, but many other driving situations, such as the ones mentioned, could also lead to false positives. It has also been acknowledged that the use of such indicators can be low, so while they can be helpful, they are not sufficient for manoeuvre prediction (Khairdoost et al., 2020). On the other hand, the minimum speed condition (signal 5) is designed to filter out city traffic events, leaving the same footprint in signals 8, 2 and 4, but that are not really overtakes. Since the trucks were operating freely in real traffic, it was not possible to enforce strict control over traffic or environmental conditions during data collection. However, the triggering rules used help ensure that the selected periods correspond to active driving, reducing the inclusion of idle, city or stop-and-go scenarios, and providing files mostly from highways or non-urban roads.

### 3.2. Machine Learning Methods

We compare three machine learning methods in this work: Artificial Neural Networks (ANN) (Haykin, 2009), Random Forest (RF) (Breiman, 2001), and Support Vector Machines (SVM, with linear and rbf kernels) (Vapnik, 1995). These are based on different classification strategies and are a popular choice in the related literature on detecting vehicle manoeuvres, as seen in Table 1 (Kim et al., 2017; Das et al., 2020, 2023) and other previous surveys (Xing et al., 2019):

- **ANN** consists of several interconnected neurons arranged in layers (i.e., input, hidden, and output layers). The nodes in one layer are interconnected to all nodes in the neighbouring (previous and subsequent) layers.

Two design parameters of ANNs are the number of intermediate layers and the number of neurons per layer.

- **RF** is an extension of the standard classification trees algorithm. It is an ensemble method where the results of many decision trees are combined. Such a combination helps to reduce overfitting and to improve generalisation capabilities. The trees in the ensemble are grown by using bootstrap samples of the data.
- **SVM** searches for an optimal hyperplane in a high-dimensional space that separates the data into two classes. Different kernel functions can be used to transform data that can be used to form the hyperplane, such as linear, or gaussian (rbf).

### 3.3. Experimental Protocol

We crop the files around the precondition trigger and use only the samples that fall within the defined crop. We crop all possible combinations of -20 seconds, -10 seconds and -5 seconds before the trigger (which we call *starttrigger*), and +0 seconds, +1 second, +2 seconds and +5 seconds after the trigger (*endtrigger*). The size of the crop (also called *time window*) has been the subject of discussion in the literature (Xing et al., 2019; Khairdoost et al., 2020). Since the existing literature has employed different crop sizes around the event of interest, here we consider several possibilities to assess the impact of its size. Afterwards, the CAN signals are processed with a sliding window of size  $w$ . Again, the many possibilities in the literature lead us to consider different sliding window sizes, in particular  $w = 0, 0.5, 1$  and  $2$  seconds, with 50% overlap. Table 3 gives the number of samples per crop considering the different combinations of *starttrigger*, *endtrigger* and sliding window sizes. For signals 1-7 (non-categorical), we compute the mean and standard deviation of the samples inside the sliding window (Kim et al., 2017), whereas for signals 8-10 (categorical), we extract the majority value among the samples. Obviously, for a sliding window size of  $w = 0$ , we take the raw samples of the file directly,

Table 3: Available samples per file and CAN channels given the different possibilities for cropping the files around the precondition trigger and the sliding window size  $w$ . Find more details in the text.

		endtrigger (seconds)																
		0				1				2				5				
		sliding window $w$ (seconds)				sliding window $w$ (seconds)				sliding window $w$ (seconds)				sliding window $w$ (seconds)				
<i>starttrigger</i> (seconds)		0	0.5	1	2	0	0.5	1	2	0	0.5	1	2	0	0.5	1	2	
		-20	201	66	39	19	211	69	41	20	221	72	43	21	251	82	49	24
		-10	101	32	19	9	111	36	21	10	121	39	23	11	151	49	29	14
		-5	51	16	9	4	61	19	11	5	71	22	13	6	101	32	19	9
		CAN channels	10	17	17	17	10	17	17	17	10	17	17	17	10	17	17	17

without any mean, standard deviation or majority computation, thus keeping the 10 CAN data channels. When  $w \neq 0$ , we effectively duplicate the number of non-categorical signals from 7 to 14, resulting in 17 CAN channels available.

All samples from overtake files are then labelled as class1 (positive class or overtake), whereas all samples for no-overtake files are labelled as class0 (negative class or no-overtake). The classifiers are trained with a portion of the files from the trucks. All other available files are used for testing. The exact number of training and test files per truck is given in Table 4. To avoid bias due to class imbalance during training, we ensure that the training dataset is balanced, containing the same number of files per truck and per class. To do so, we check how many files of each class are available per truck, then we take 70% of the minimum. The test set is imbalanced, which we address by reporting results using AUC-ROC (Area Under the Receiver Operating Characteristic Curve), which is robust to class imbalance, and by providing TPR and TNR separately rather than using global accuracy. Experiments are conducted using Matlab r2023b. All classifiers are left with the default values (ANN: one hidden layer with 10 neurons; RF: 100 decision trees), except:

- ANN and SVM: we use standardisation in non-categorical signals by sub-

Table 4: Files employed for training and test per class and per truck given the different possibilities for cropping the files around the precondition trigger.

truck	<i>starttrigger</i>	training (both classes)				test class0 (no overtake)				test class1 (overtake)			
		<i>endtrigger</i>				<i>endtrigger</i>				<i>endtrigger</i>			
		0	1	2	5	0	0.5	1	2	0	0.5	1	2
t1	-20	72	72	72	68	35	35	34	33	325	325	324	328
	-10	74	74	74	70	36	36	35	34	336	336	335	339
	-5	79	79	78	74	38	37	37	35	333	333	333	337
t2	-20	37	37	37	37	117	117	117	117	43	43	43	43
	-10	38	38	38	38	119	119	119	119	45	45	45	45
	-5	38	38	38	38	121	121	121	121	45	45	45	45
t3	-20	4	4	4	4	2	2	2	2	7	7	7	7
	-10	4	4	4	4	2	2	2	2	7	7	7	7
	-5	4	4	4	4	3	3	3	3	7	7	7	7
t4	-20	46	45	44	42	21	20	19	18	259	260	261	263
	-10	50	49	47	45	22	21	21	20	270	271	273	275
	-5	52	51	49	47	23	22	22	21	282	283	285	287
t5	-20	3	3	3	3	2	2	2	2	8	8	8	8
	-10	3	3	3	3	2	2	2	2	8	8	8	8
	-5	3	3	3	3	2	2	2	2	8	8	8	8

tracting the mean and dividing by the standard deviation of the training samples

- The ANN iteration limit is raised to 1e6 (from 1e3) and the SVMrbf iteration limit is raised to 1e8 (from 1e6) to facilitate convergence

In addition, all classifiers are set up to provide class posterior probabilities as output. Thus, their output is a continuous value in [0, 1]. If the output is higher than 0.5, the input is classified to belong to the positive class, and to the negative class if lower than 0.5.

#### 4. Results

To keep consistency with our previous work (Butt et al., 2024), we start the analysis only with trucks t1-t3. This is because data of these trucks was the

only available data when this research started, in particular when we initially trained the classifiers with different combinations of *starttrigger*, *endtrigger* and sliding window sizes. Later on, when data from trucks t4-t5 was made available, we tested the classifiers trained with t1-t3 and also considered retraining the classifiers with data from all trucks.

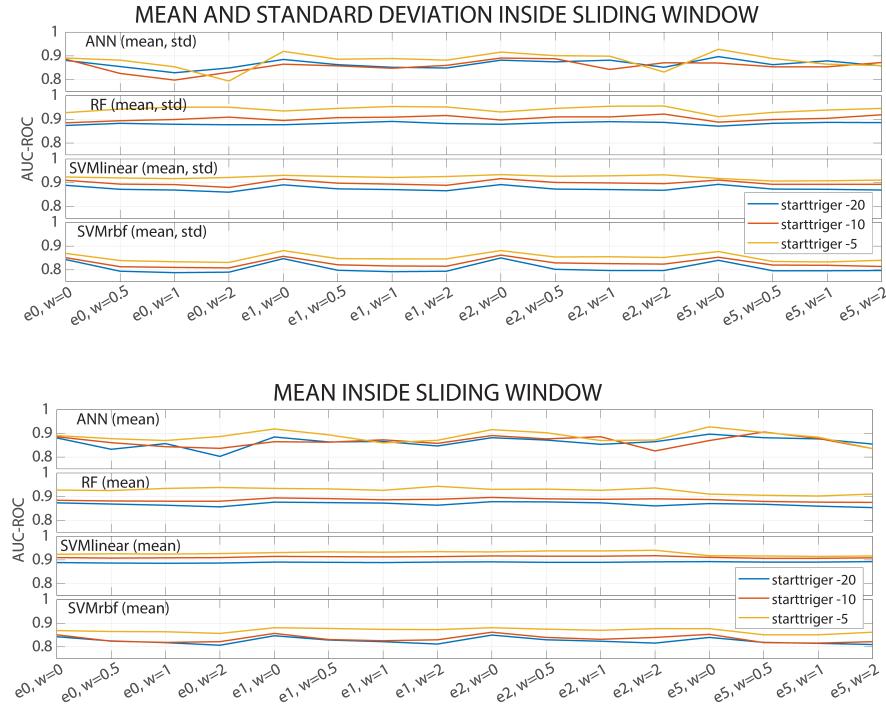


Figure 2: AUC-ROC values for trucks t1-t3 (*mirror* trucks) when the classifiers are trained with data from trucks t1-t3. Results are given for different combinations of crops before the precondition trigger (*starttrigger* in the legends), after the precondition trigger (parameter  $e$  in the horizontal axes) and sliding window sizes (parameter  $w$ ). Better seen in colour.

#### 4.1. Parameter analysis of the classifiers

We first analyse (Figure 2) the performance of *mirror* trucks (t1-t3) in terms of the AUC-ROC of the classifiers, considering the values of *starttrigger*, *endtrigger*, and sliding window sizes tested in this study. We also distinguish between

computing both the mean and standard deviation of non-categorical signals within the sliding window (top) or only the mean (bottom). The AUC-ROC summarises the ability of a model to discriminate between positive and negative instances across all classification thresholds. The AUC-ROC score ranges from 0 to 1, where 0.5 indicates random guessing, and 1 indicates perfect classification performance. As can be observed, the AUC-ROC is inversely correlated in the majority of cases with the length of *starttrigger*, i.e. the closer to the actual overtaking event, the better the accuracy. This is reflected by the orange curve (*starttrigger*=-5) being above the red and blue ones (*starttrigger*=-10/-20). A larger *starttrigger* crop may introduce unnecessary data too early in time that can confuse the models, reducing their ability to discriminate between overtaking and non-overtaking scenarios. On the other hand, this highlights the potential challenge of detecting the event of interest well in advance of its onset.

Given the superior performance of *starttrigger*=-5, we focus on this configuration and include as metrics the True Positive Rate (*TPR*) and True Negative Rate (*TNR*). The *TPR* measures the proportion of positive instances (overtakes) correctly classified as positives, while the *TNR* (also referred to as specificity) indicates the proportion of negative instances (no-overtakes) correctly classified as negatives. *TPR* is calculated as  $TPR = TP / (TP + FN)$ , where *TP* is the number of true positives, and *FN* is the number of false negatives. Similarly, *TNR* is calculated as  $TNR = TN / (TN + FP)$ , where *TN* is the number of true negatives, and *FP* is the number of false positives. Figure 3 (top row) presents the results for the *mirror* trucks.

The first observation is that both *TNR* and *TPR* appear relatively balanced across all classifiers. That is, for each classifier, both metrics are generally within similar ranges, with only slight variations. When examining *TNR*, Random Forest (red curves) outperforms the other classifiers, followed by SVM linear (green). In contrast, ANN (blue) and SVM rbf (black) are the worst, with their *TNR* in similar ranges. For *TPR*, Random Forest and SVM linear show comparable accuracy, followed by ANN and then SVM rbf. Another observation is that using only the mean (dashed curves) vs. both the mean and standard

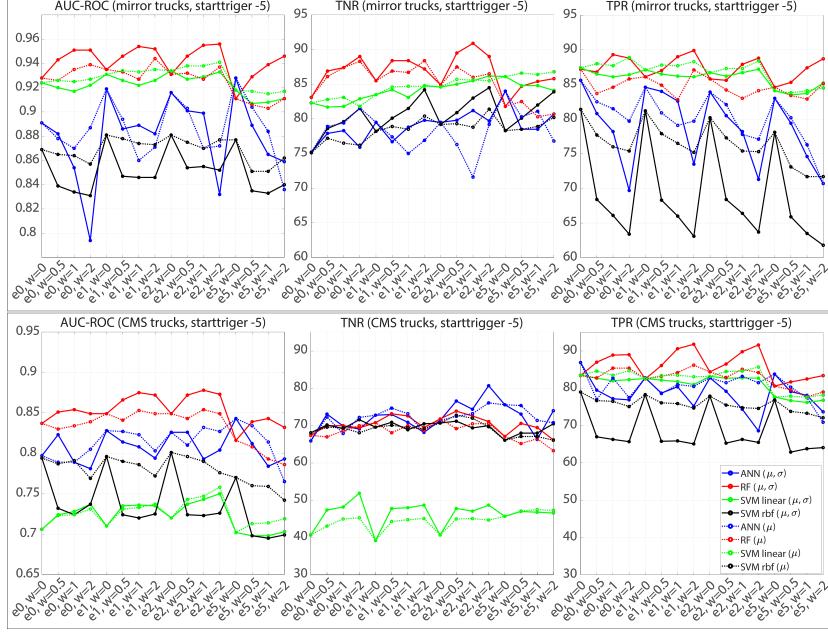


Figure 3: AUC-ROC, TNR and TPR values for trucks t1-t3 (top, *mirror trucks*) and t4-t5 (bottom, *CMS trucks*) when the classifiers are trained with data from trucks t1-t3. The plot uses a value of *starttrigger*=-5 (the best case of Figure 2). Solid lines indicate that the mean and standard deviation of non-categorical signals are computed inside the sliding window, whereas dashed lines indicate that only the mean is computed. ex in the x-axes denotes the length of *endtrigger* (given by x), and *w* denotes the sliding window size. Better seen in colour.

deviation (solid curves) is mostly irrelevant for *TNR*, but *TPR* is more sensitive. The solid curves, which incorporate both the mean and standard deviation, have greater *TPR* variability, being dependent on the size of the sliding window. In addition, which option is the best depends on the classifier. RF performs better when using both the mean and standard deviation (solid red curve), whereas the other classifiers prefer the mean only (dashed curves).

Interestingly, the optimal sliding window size varies between classifiers too. Both ANN and SVMs perform best with *w*=0 (no sliding window, or using the raw samples), making the discussion of mean vs. both mean and standard

deviation irrelevant. On the other hand, Random Forest shows the best performance with larger window sizes, specifically  $w=1$  or 2. Regarding the impact of *endtrigger* (the end of the crop after the precondition trigger), we observe that *TNR* benefits from a larger crop, with curves increasing towards the right of the  $x$ -axis. A larger crop provides more context and allows the model to confirm that no overtaking is taking place but another manoeuvre instead, reducing false positives. Conversely, *TPR* has an opposite trend, suggesting that the most useful information for detecting the overtaking class disappears shortly after the trigger. As we will see later (Section 4.4 and second column of Figures 5-6), the dynamics of the CAN signals are different before and after an overtaking starts. As a result, extending the crop too far beyond the trigger introduces unnecessary data that can lead to an increase in false negatives.

#### 4.2. Comparison of mirror and CMS trucks

We then analyse the performance of the classifiers on *CMS* trucks (t4-t5). Results are given in the bottom row of Figure 3. Recall that the algorithms are still trained on data from *mirror* trucks (t1-t3) only, so in this sub-section, we are evaluating how the classifiers behave when tested on trucks that differ from the training set.

Comparing the top and bottom rows of Figure 3, we observe that *TPR* remains consistent across truck types for all classifiers. In some cases, it even improves, e.g. the best performance of RF exceeds 90%, or the variability of ANN and SVM rbf with respect to  $w$  is less pronounced. These findings suggest that the ‘execution’ of an overtaking remains largely invariant, so the *TPR* is not significantly impacted when trained and tested on overtakes from different trucks or drivers. However, *TNR* shows a significant decline in *CMS* trucks, which significantly affects the AUC-ROC. The drop is particularly pronounced for SVM linear, which goes from having the second-best *TNR* to performing worse than a random guess. This suggests that the patterns of no-overtake signals differ between mirror and CMS datasets, a phenomenon that we will further analyse in Section 4.4.

The majority of observations regarding the optimal sliding window size or the impact of *endtrigger* also hold in this sub-section. ANN and SVMs perform best with  $w=0$  (at least in terms of *TPR*), while RF does better with larger windows ( $w=1$  or 2). Similarly, *TPR* worsens if the end crop is extended too far beyond the precondition trigger. Additionally, RF prefers the use of mean and standard deviation as features, whereas the other classifiers perform better with the mean only.

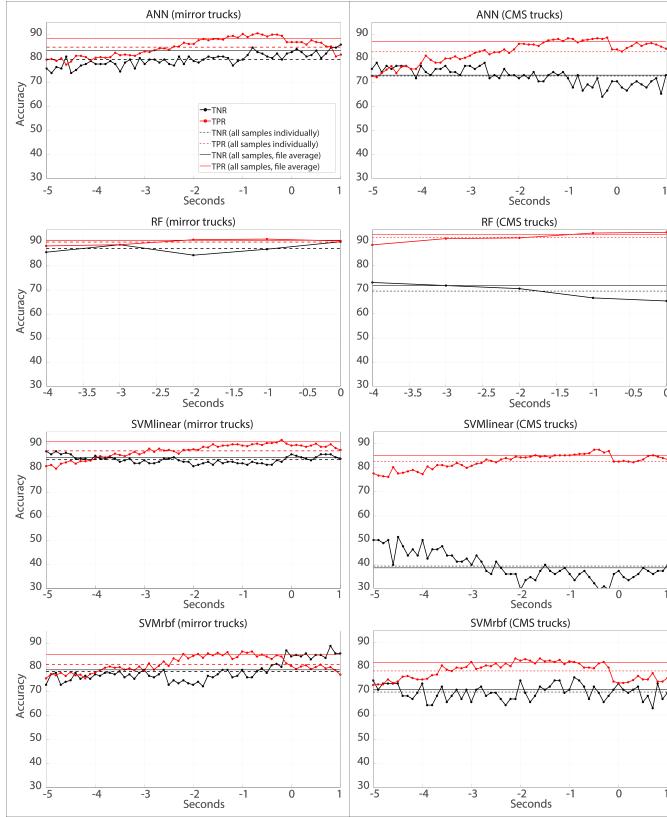


Figure 4: TNR/TPR of the classifiers at different moments before/after the precondition trigger for trucks t1-t3 (left, *mirror* trucks) and t4-t5 (right, *CMS* trucks) when the classifiers are trained with data from t1-t3. See Section 4.3 for more details about the configuration of the experiments. Better seen in colour.

#### 4.3. Prediction window analysis

From the results of previous sub-sections, we carry forward the following best configurations for further analysis:  $starttrigger=-5$ ,  $endtrigger=1$  (a compromise between a larger crop for higher  $TNR$  vs. a shorter crop for higher  $TPR$ ), and  $w=0$  for all classifiers (except  $w=2$  for RF, which is applied using mean and standard deviation within the sliding window). We then analyse in this sub-section the  $TNR/TPR$  at different time points around the precondition trigger (Figure 4). To obtain these plots,  $TNR/TPR$  are computed using only the samples corresponding to each specific time point. The classifiers remain trained exclusively on data from t1-t3. The following additional results are also shown in the figure:

1. Per-sample  $TNR/TPR$  (dashed horizontal lines), computed using all samples from each file individually. This generates one decision score per sample of the file, corresponding to how accuracy metrics were previously calculated for Figures 2 and 3.
2. Per-file  $TNR/TPR$  (solid horizontal lines), computed by averaging the decision scores of all samples within a file. This produces one decision score per file, integrating the output of the classifiers for all samples in the time crop.

With *mirror* trucks (t1-t3), left column, the accuracy (both  $TNR$  and  $TPR$ ) is observed to improve as we approach the trigger. After the trigger at  $x=0$ ,  $TPR$  (red curves) decreases. This is consistent with the previous observation that a large  $endtrigger$  value was detrimental to  $TPR$ . The findings of this subsection confirm that the most relevant information for the overtaking class is contained before the trigger. The curves also confirm the previous observation that  $TNR/TPR$  are balanced with these trucks, i.e. they are in the same range for most classifiers, with differences of 5-10% maximum at some time points.

It can also be seen that the solid horizontal lines (integrating the scores of the entire crop) are a good option to obtain high accuracy, counteracting the

variations of accuracy with time. The downside is that it needs to wait until all samples of the crop are available to make a decision. However, with a classifier like RF, both  $TNR$  and  $TPR$  are close to or higher than 90% at all times, highlighting the power of this classifier for our task.

With  $CMS$  trucks (t4-t5), right column, the same trend is observed for  $TPR$ , i.e. it increases as we approach the trigger. However,  $TNR$  actually decreases towards the trigger. Also, as observed in the previous sub-section,  $TNR$  with  $CMS$  trucks is much lower than  $TPR$ . We attribute this to a difference in the patterns of no-overtake between the two trucks' sets, which we will analyse in the following sub-section.

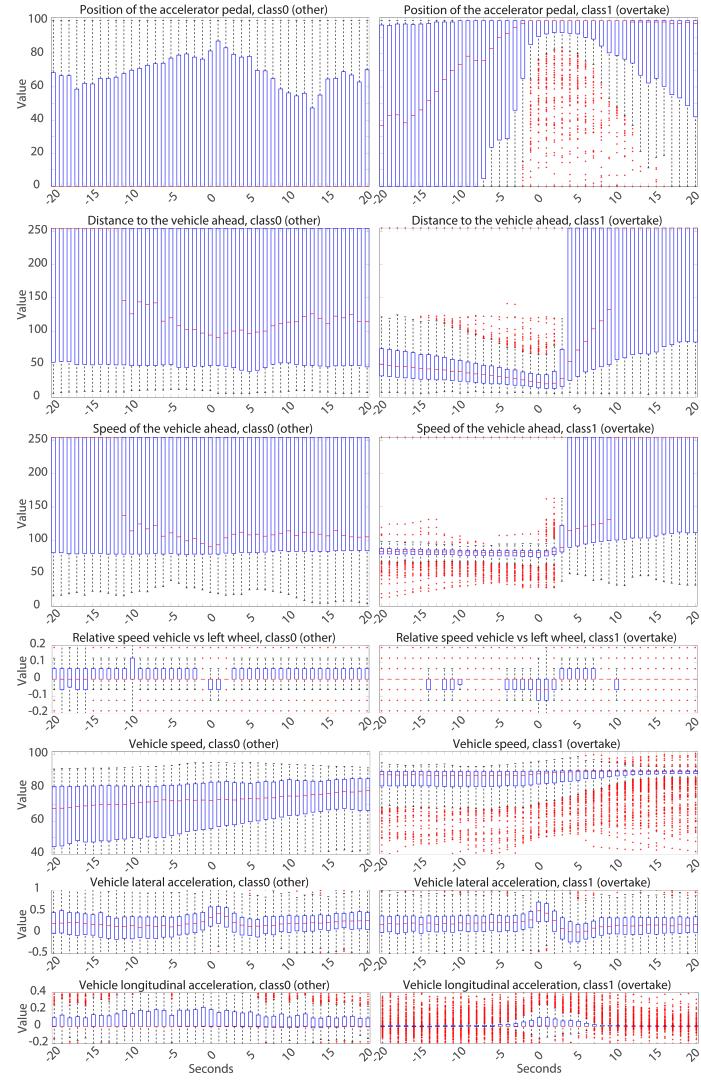


Figure 5: Mirror trucks (t1-t3): boxplots of the CAN signals  $\pm 20$  seconds around the precondition trigger. Left column: no-overtake class. Right: overtake class.  
Better seen in colour.

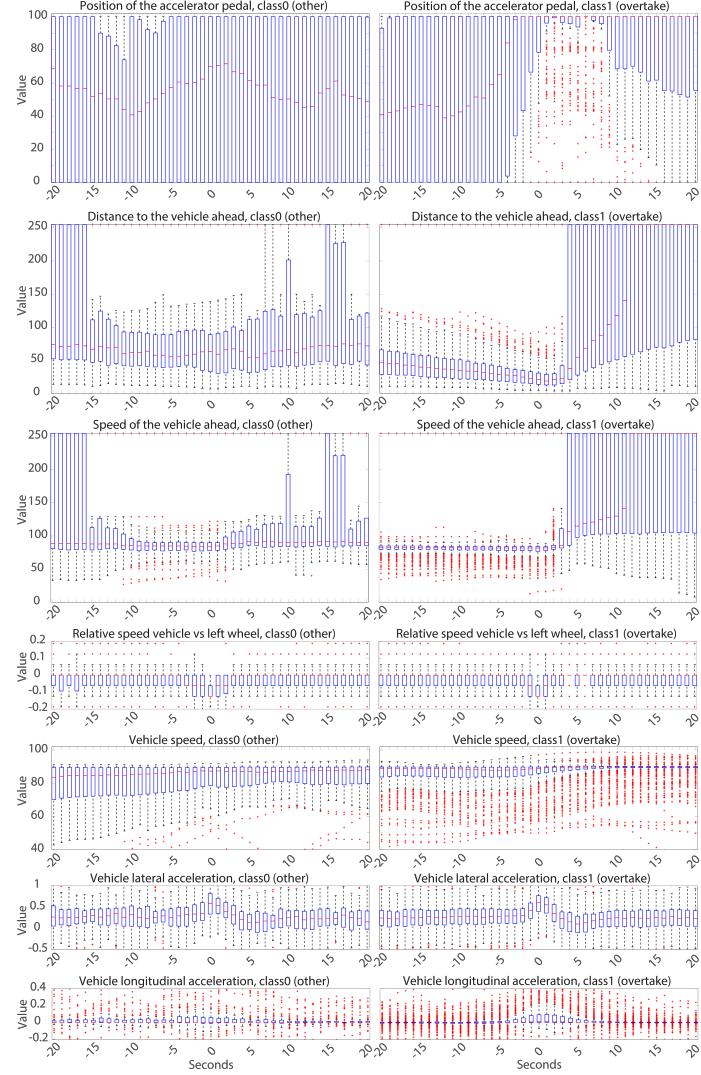


Figure 6: CMS trucks (t4-t5): boxplots of the CAN signals  $\pm 20$  seconds around the precondition trigger. Left column: no-overtake class. Right: overtake class. Better seen in colour.

#### 4.4. CAN signals analysis

In this sub-section, we directly examine the patterns of the CAN signals to understand the dynamics and behaviour of the drivers in an overtake. It will also provide insights into the observed performance differences between t1-t3

(*mirror* trucks) and t4-t5 (*CMS* trucks). We do the analysis by plotting in Figures 5 and 6 the boxplots of the CAN signals of each class for the two truck groups over the interval from -20 to +20 seconds relative to the precondition trigger.

We begin by analyzing the CAN signals of *mirror* trucks (t1-t3), Figure 5, focusing on each signal individually:

- *Position of the accelerator pedal.* During overtakes, the accelerator is increasingly pushed, often exceeding 90% as the trigger approaches. After the trigger, some drivers release the pedal slightly, but most maintain high pressure. In no-overtake scenarios, the pedal position ranges from not pushed to moderately pushed, likely indicating the use of cruise control in many cases, since the median is always at 0.
- *Distance to the vehicle ahead.* In overtakes, the median distance steadily decreases below 50m before abruptly jumping to 255m after the trigger, signalling a lane change (no vehicle ahead). Occasionally, other vehicles are detected beyond 100m after the trigger, suggesting that the new lane is not empty. The majority of lane changes occur around 25m from the vehicle ahead, though some happen at greater distances. In no-overtakes, the median distance remains consistently high, around 100-125m, with occasional values at 255m, indicating no vehicle ahead.
- *Speed of the vehicle ahead.* This signal behaves in a similar way to the previous one, with minor differences. In overtakes, the vehicle ahead maintains a constant median speed, typically around 90 km/h before the trigger, consistent with either another truck (speed-limited by software) or a road limit. After the trigger, the signal then jumps abruptly. Segments with a median of 120 km/h likely correspond to motorways.
- *Relative speed vehicle/wheel.* In overtakes, this signal activates primarily around the trigger. It initially takes negative values (wheel turned left), then rises to positive values (wheel turned right, returning to straight),

and eventually stabilises at zero (truck moving straight). In no-overtake scenarios, the signal is mostly zero or positive, indicative of other manoeuvres.

- *Vehicle speed.* During overtakes, the median speed remains close to 90 km/h, with minimal variability, indicating the truck is near its maximum speed before overtaking. In no-overtakes, speed is more variable, with a lower median and a slight increase over time.
- *Vehicle lateral acceleration.* In both classes, lateral acceleration is positive due to the leftward motion that triggers the precondition. However, in overtakes, a sharper acceleration increase is observed at the trigger, followed by a pronounced decrease or even negative values afterwards. This suggests a lane change, distinguishing it from other manoeuvres like turning left.
- *Vehicle longitudinal acceleration.* In overtakes, longitudinal acceleration is minimal, as the truck is already moving at high speed in most cases. A few instances show slight acceleration at the trigger, very likely those which were seen to increase speed after the trigger. At the end of an overtake, a deceleration phase (negative values) might be expected if the truck slows down to adapt its speed when returning to the right lane. However, the absence of such deceleration in the plot suggests that, in the analysed segments, overtakes extend beyond the 20-second time window shown in the  $x$ -axis. In no-overtakes, the observed gradual speed increase explains the consistently positive acceleration values.

These differences between classes suggest that it is feasible to distinguish them through classifier training. We now examine the signals of the *CMS* group ( $t_4-t_5$ ) in Figure 6, comparing them with the signals of *mirror* trucks analyzed above:

- *Position of the accelerator pedal.* In the no-overtake class, the signal shows higher pedal positions, often exceeding 50%, whereas *mirror* trucks ex-

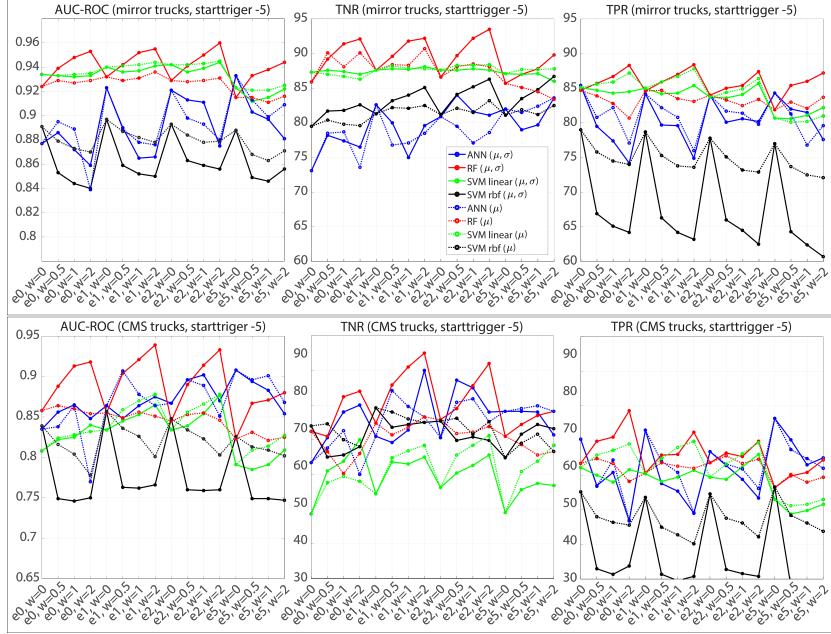


Figure 7: AUC-ROC, TNR and TPR values for trucks t1-t3 (top, *mirror* trucks) and t4-t5 (bottom, *CMS* trucks) when the classifiers are trained with data from trucks t1-t5. The plot uses a value of *starttrigger*=-5 (the best case of Figure 2). Solid lines indicate that the mean and standard deviation of non-categorical signals are computed inside the sliding window, whereas dashed lines indicate that only the mean is computed. Better seen in colour.

hibited low values in this scenario. This indicates that in the *CMS* set, drivers were not relying on cruise control as frequently during no-overtake manoeuvres, but manually managing the truck speed, so the status of the cruise control does not appear to be a reliable separator of the two classes under analysis. As argued below, no-overtake *CMS* data has probably been collected under heavier data traffic, not allowing for much cruise control usage. The overtake class, on the other hand, exhibits a consistent pattern across both truck groups.

- *Distance to the vehicle ahead.* In no-overtake manoeuvres, the signal shows a lower tendency to reach 255 (indicative of no vehicle ahead). This sug-

gests that *CMS* data was more likely collected under heavier traffic conditions. In contrast, *mirror* trucks generally maintained higher distances or had no vehicle ahead more frequently. In overtaking manoeuvres, on the other hand, both truck types exhibit similar patterns, with distances decreasing as the vehicle approaches, followed by a signal jump when the overtake is initiated, indicating no vehicle ahead.

- *Speed of the vehicle ahead.* This signal reflects the same differences as observed in the distance signal, reaffirming that *CMS* trucks were operating in potentially denser traffic environments.
- *Relative speed vehicle/wheel.* This signal exhibits relatively similar patterns between both classes in *CMS* trucks, contrasting with the greater variability observed in *mirror* trucks. This could reflect the random nature of the real-world driving data collection, potentially linked to conditions such as the heavier traffic noted. What remains consistent in overtakes is a distinct decrease towards more negative values near the trigger (indicating the wheel turned left), followed by a return to near-zero values during a few seconds (return to wheel straight).
- *Vehicle speed.* The pattern of this signal is similar to the *mirror* trucks, with a slightly higher median during no-overtakes.
- *Vehicle lateral acceleration.* Patterns of lateral acceleration are comparable between *CMS* and *mirror* trucks, indicating similar dynamics during overtakes or other lateral manoeuvres.
- *Vehicle longitudinal acceleration.* While the general trend is similar to *mirror* trucks, *CMS* trucks exhibit less variability in acceleration during no-overtakes. This aligns with the higher median speed observed earlier, reducing the need for acceleration.

In summary, while many signal patterns remain consistent across truck groups, key differences in no-overtake manoeuvres (particularly in speed, pedal

position, and distance signals) suggest different driving contexts, possibly influenced by traffic density, transport task, and driving style variations. This could explain the variations in performance between the two truck groups since the classifiers were trained with data from one group only. The observed differences affect mostly the no-overtake class, which justifies the decrease in  $TNR$  since the no-overtake patterns of the *CMS* set have not been seen yet by the classifiers. On the other hand, the behaviour of drivers during overtakes appears to be more standardised, which explains that  $TPR$  is high and consistent across truck groups. Signals such as distance to the vehicle ahead, speed of the vehicle ahead, and lateral acceleration follow similar patterns when the trucks approach the vehicle ahead. The signals then exhibit characteristics indicative of a lane change to the left, with no vehicle ahead or at a high distance.

#### *4.5. Retraining with all trucks*

To counteract the observed differences in the CAN signals of the non-overtake class, we retrain the classifiers using data from all trucks. Figure 7 shows the  $AUC$ ,  $TNR$  and  $TPR$  in the same way as Figure 3. It can be seen that the  $TNR$  of both truck types generally increases when training with data from all trucks. Thus, incorporating the different observed patterns of no-overtakes improves the classifier's ability to detect them. However,  $TPR$  is lower for both truck types. Since the overtake patterns appear similar between truck types based on the boxplots, the increased variability in non-overtake signals seems to impact overtake classification negatively. One possible interpretation is that expanding the no-overtake feature space brings it closer to the overtake feature space, causing the classifiers to shift the decision boundary toward the overtake samples, thus increasing misclassification. Other phenomena observed earlier in Figure 3 also apply here, such as the continued superiority of Random Forest, its preference for larger  $w$  values, and the inclusion of both the mean and standard deviation of the signals. Meanwhile, other classifiers still tend to perform better with  $w=0$ .

Figure 8 then shows the  $TNR/TPR$  at different moments around the pre-

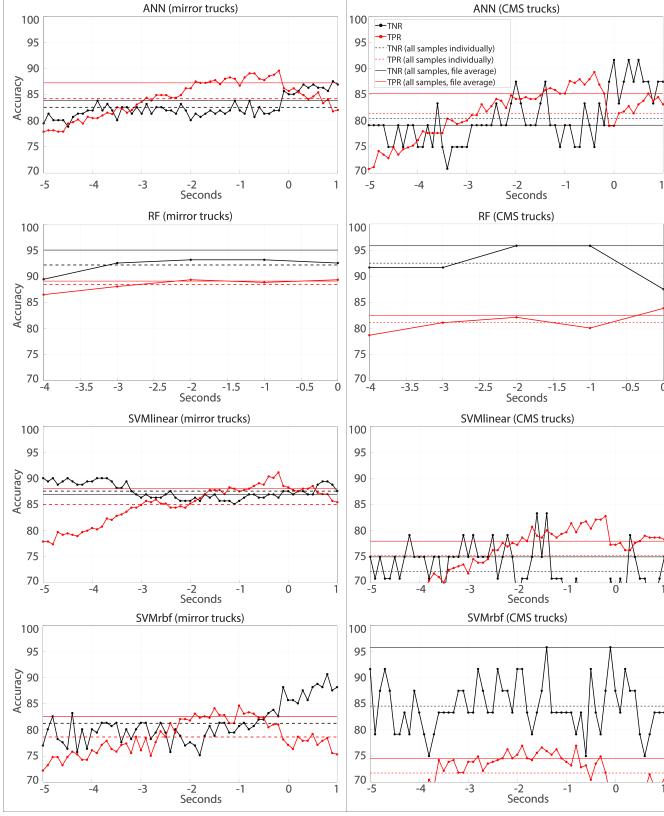


Figure 8: TNR/TPR of the classifiers at different moments before/after the precondition trigger for trucks t1-t3 (left, *mirror* trucks) and t4-t5 (right, *CMS* trucks) when the classifiers are trained with data from t1-t5. See Section 4.5 for more details about the configuration of the experiments. Better seen in colour.

condition trigger, in the same way as Figure 4. A key observation is that now, the *TNR/TPR* of both truck types tend to remain steady or increase towards the trigger, with values mostly above 70% in the entire crop (notice the change of range in the *y*-axes w.r.t. Figure 4). Additionally, the difference between *TNR/TPR* is now less pronounced. As in Figure 4, *TPR* (red curves) decreases after the trigger, indicating that the most important information for the overtake class is located before the trigger.

The solid horizontal lines (which integrate the scores across the entire crop)

Table 5:  $TNR/TPR$  of the classifiers comparing the two training possibilities. Results are given considering a single classification score, computed as the average of scores within the crop.

classifier	train	test			
		t1-t3 ( <i>mirror</i> )	t4-t5 ( <i>CMS</i> )	TNR	TPR
ANN	t1-t3	83.2	<b>88.3</b>	73.1	<b>87.0</b>
	t1-t5	<b>83.9</b>	87.3	<b>79.2</b>	85.2
RF	t1-t3	90.7	<b>90.6</b>	71.8	<b>93.0</b>
	t1-t5	<b>95.0</b>	89.1	<b>95.8</b>	82.5
SVM linear	t1-t3	84.5	<b>90.9</b>	38.5	<b>85.2</b>
	t1-t5	<b>87.0</b>	88.0	<b>75.0</b>	78.0
SVM rbf	t1-t3	78.9	<b>85.2</b>	70.5	<b>81.7</b>
	t1-t5	<b>82.6</b>	82.6	<b>95.8</b>	74.6

are also seen to offer an optimal solution, counteracting the oscillations of accuracy over time. We give in Table 5 such integrative  $TNR/TPR$  values depicted in Figures 8 and 4 with solid lines. As observed earlier,  $TNR$  improves for all truck types when training with data from all trucks, whereas  $TPR$  decreases. However, the gains in  $TNR$  (by 25% or more with certain classifiers) are more substantial than the decrease in  $TPR$  (which is at most 10.5%, and typically 2-3%).

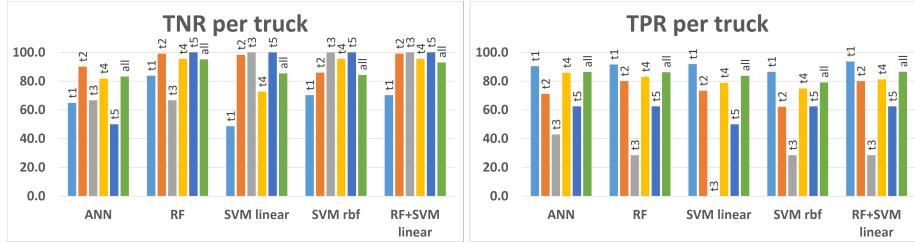


Figure 9:  $TNR$  and  $TPR$  values of the classifiers per truck when the classifiers are trained with data from trucks t1-t5. Better seen in colour.

#### 4.6. Per-truck analysis and classifier fusion

We finally show the performance per truck, considering the configuration of the classifiers of the last sub-section (trained with both *mirror* and *CMS*

Table 6: *TNR* and *TPR* values of the classifiers per truck when the classifiers are trained with data from trucks t1-t5. The best cases per column are marked in bold.

classifier	TNR						TPR					
	t1	t2	t3	t4	t5	all	t1	t2	t3	t4	t5	all
ANN	64.9	90.1	66.7	81.8	50.0	83.2	90.4	71.1	<b>42.9</b>	<b>85.9</b>	<b>62.5</b>	86.4
RF	<b>83.8</b>	<b>99.2</b>	66.7	<b>95.5</b>	<b>100.0</b>	<b>95.1</b>	91.6	<b>80.0</b>	28.6	83.0	<b>62.5</b>	86.2
SVM linear	48.6	98.3	<b>100.0</b>	72.7	<b>100.0</b>	85.4	91.9	73.3	0.0	78.8	50.0	83.7
SVM rbf	70.3	86.0	<b>100.0</b>	<b>95.5</b>	<b>100.0</b>	84.3	86.5	62.2	28.6	74.9	<b>62.5</b>	79.1
RF + SVM linear	70.3	<b>99.2</b>	<b>100.0</b>	<b>95.5</b>	<b>100.0</b>	93.0	<b>93.7</b>	<b>80.0</b>	28.6	81.3	<b>62.5</b>	<b>86.5</b>

trucks). Results are given in Figure 9 and Table 6.

As can be seen, the performance varies between trucks. *TPR* shows a consistent relative tendency across all classifiers, with t1 performing the best, followed by t4, t2, t3, and t5. Notice that this performance trend correlates with the size of the overtaking class in the training/test sets (Table 4). Despite the classifiers being trained with balanced classes, t3 and t5 just have 3-4 training files per truck. This smaller sample size for these trucks may mean that the classifiers are not sufficiently trained on them. Although driver or truck model information is not available, each truck is likely driven by a different person. Even if overtaking patterns were observed to be consistent (Figures 5 and 6), small individual differences per driver or truck type may occur, which could impact the *TPR* given the smaller training sample size for t3 and t5.

*TNR*, on the other hand, is not consistent across classifiers and trucks. For example, t3 has a poor *TNR* with RF and a good *TNR* with SVM linear, while t4 shows the opposite pattern. This inconsistency may arise from the heterogeneous nature of the no-overtaking class. Our dataset includes a wide range of driving situations in such class, such as turns, avoiding an exit-only lane, giving way to a vehicle, surpassing stationary vehicles, and others. However, we have not systematically captured these additional possibilities, as the dataset comes from real-world trucks in regular operation, where such events can occur randomly. An obvious solution would be to expand the dataset, thereby improving the representativeness of the various non-overtaking manoeuvres.

Given such different  $TNR$  performance per classifier and per truck, we tried classifier fusion by averaging their output scores. We tested all possible combinations of 2, 3 and 4 classifiers. The most consistent was observed to be Random Forest combined with SVM linear, whose results are provided in Figure 9 and Table 6 too. As shown in the plot, the fusion boosts  $TNR$  of most trucks, with the exception of t1, which experiences a decline.  $TPR$ , on the other hand, does not improve with fusion (except for t1, where  $TNR$  worsened). Overall, while the fusion does not impact  $TPR$ , it helps to balance and improve  $TNR$ , and as observed in the table, the fusion yields the best performance across most trucks.

## 5. Conclusions

Safe truck overtakes are crucial to avoid collisions, reduce congestion, and guarantee smooth traffic flow. Their reliable prediction can enable advanced Driver Assistance Systems (ADAS) to provide timely, informed driving decisions, ultimately enhancing road safety and efficiency. This study investigates the capabilities of machine learning classifiers trained on CAN (Controller Area Network) signals to detect overtakes in trucks. In particular, we have applied traditional, widely used classifiers in related tasks on vehicle manoeuvre detection (Xing et al., 2019; Kim et al., 2017; Das et al., 2020), such as Support Vector Machines (SVM), Random Forest (RF), and Artificial Neural Networks (ANN). For this study, we gathered CAN data from actual operating trucks provided by the Volvo Group participating in this research. To the best of our knowledge, we are among the first to address overtaking detection in trucks, particularly using real CAN bus data.

To collect data, we designed a precondition trigger representative of an overtaking, so the data logger only records when a certain combination of CAN signals occurs. This prevented storage problems and allowed us to obtain segments where a possible overtaking occurs. Data from a dashcam was also captured, which allowed offline manual labelling of the obtained segments. With this procedure, we obtained 865 overtaking segments and 382 no-overtaking segments from

5 different trucks. The precondition trigger was designed to filter out city traffic events (below 50 km/h) that are not really overtakes, or at least not as dangerous as an overtake at high speeds, so the resulting files mostly correspond to highways or non-urban roads.

Throughout different experiments, we then demonstrate the suitability of CAN bus data to detect overtakes in trucks. We have examined a variety of configurations (Figure 1) that involve, for example, cropping the files around the precondition trigger with different time lengths, processing CAN signals with a sliding window of different lengths, or extracting various metrics within the window, such as the mean or the standard deviation of the signals. This is because the value of such parameters is subject to discussion in the related literature (Xing et al., 2019; Khairdoost et al., 2020). Relevant findings include, for example, that taking a very large crop before or after the trigger is detrimental to detecting overtakes. Our optimal segment crop comprises 5 seconds before the trigger and just 1 second afterwards. Thus, the most helpful information for detecting the overtake class becomes more evident a few seconds before the trigger and disappears shortly after. This somehow limits the capability of the system to detect overtakes reliably to just a few seconds before it starts.

Other findings concern the optimal sliding window size or the metrics to be gathered within the window, which is observed to be different per classifier. We have also analysed the pattern of CAN signals to understand their dynamics. The behaviour of drivers during overtakes appears to be consistent across different trucks, with many signals remaining consistent. Two of the trucks have a mirrorless Camera Monitoring System (CMS), while the other three have a standard physical mirror. Although it was not our primary aim in this paper to compare them, we have not observed any remarkable difference in the pattern of overtake signals between the two truck types, suggesting that the absence of a physical mirror does not substantially affect the way that drivers carry out an overtake. On the other hand, differences are observed in no-overtake manoeuvres. Data from some files is observed to be collected under heavier traffic conditions than others, which results in a different pattern of CAN sig-

nals. However, we have not systematically controlled such scenarios. Our data was captured with actual operating trucks normally driven, so the driving moments where the trigger activates can occur randomly. The effect is that while the Positive Rate of the true class ( $TPR$ ) remains consistent, the True Negative Rate ( $TNR$ ) is highly dependent on the presence of different non-overtake patterns in the training data.

In the last part of our study, we analysed performance at the individual truck level. For the overtaking class, we observed a high dependency on the amount of training data available per truck, regardless of the classifier. This suggests that subtle differences per driver or truck type may also introduce inconsistencies in the training set if they are not adequately represented. On the contrary, the no-overtake class exhibited less consistency. We did not observe a clear dependency on the size of the training set or the classifier type. We believe that the non-systematic nature of the data acquisition process mentioned earlier may produce such a lack of consistency. An obvious approach would be to increase the database, considering sufficient representativity of possible non-overtake manoeuvres. Another way that we have found to counteract such an effect is classifier fusion. Through simple score fusion of two classifiers (RF and SVM linear), the  $TNR$  of most trucks is boosted to more than 95%, without the  $TPR$  being significantly impacted. Overall, the fusion provides a  $TNR$  of 93% for all trucks, and a  $TPR$  of 86.5%.

### 5.1. Limitations and Future Work

In the context of ADAS, both false positives (related to  $TNR < 100\%$ ) and false negatives ( $TPR < 100\%$ ) carry practical safety implications. The potential impact of these errors depends on how the detection module is integrated into the ADAS pipeline. For real-time systems, false positives can be particularly problematic, since erroneous alerts may distract or confuse the driver, especially if they happen repeatedly. If the system is designed to adapt vehicle behaviour (e.g., adjusting speed, spacing, or lane position), it could cause unwarranted actions, potentially increasing risk instead of reducing it. Ultimately, false pos-

itives can erode driver trust, leading to disengagement or disregard of future valid alerts. In contrast, false negatives (undetected overtakes) in real-time reduce the system’s ability to provide timely support or warnings. It may be acceptable in non-critical situations or when the driver is in full control already. However, while less disruptive than false positives, frequent false negatives may affect the usefulness of the system in critical situations or degrade the coverage of the ADAS functionality. On the other hand, in offline analysis, such as post-trip analysis, driver profiling, or fleet safety auditing, classification errors have less immediate impact. False positives typically lead to the mislabeling of routine driving as overtaking, slightly skewing statistics but without real-time consequences. False negatives reduce completeness but rarely compromise safety. Nonetheless, systematic errors or biases can still distort behavioural summaries or affect managerial decision-making. Therefore, maintaining high reliability remains important even in non-real-time pipelines.

In terms of real-time feasibility, although not directly evaluated in this study, the methods used are based on lightweight models (ANN, RF, SVM) and low-dimensional CAN signal statistics, which are well-suited for embedded implementation. In future work, we plan to port the trained classifiers to onboard platforms and evaluate inference times and resource consumption. Given the simplicity of the models and the preprocessing pipeline, we expect that real-time performance can be achieved with minimal computational overhead. In addition, all three classifiers can be trained efficiently due to moderate input dimensionality. RF trains within seconds. SVM, though more computationally demanding, keeps training times between seconds and minutes on a standard workstation for linear kernels. The ANN used has a simple architecture (one hidden layer), and training converges in less than one minute. Beyond technical performance, our system has managerial and industrial relevance. Since it relies solely on CAN bus signals, it can be deployed without requiring additional sensors, making it a cost-effective, scalable, and privacy-preserving solution suitable for fleet-wide deployment. In addition to potential real-time safety applications, the system can support long-term driver behaviour monitoring

and profiling, safety auditing, insurance risk assessment, and driver training programs by identifying risky patterns or habits.

Several factors may contribute to classification errors, including variability in traffic context, differences in driving styles, sensor noise, and the limited number of labelled overtaking events, especially in some trucks. These factors may result in signal patterns that deviate from typical examples seen during training. To mitigate these risks and improve the reliability of the system in operational settings, future implementations may incorporate more sophisticated temporal smoothing techniques (e.g. decision aggregation over consecutive windows of shorter or longer duration, that also go earlier in time), combined with variable confidence-based thresholds, so the evidence is accumulated over time as the overtaking event approaches, rather than relying on the output of a single crop. Hybrid sensor fusion could be another possibility, combining CAN-based predictions with camera or radar inputs, although it goes beyond our purpose of employing only CAN signals.

The above experiments have been carried out with the majority of the parameters of the classifiers set to default. Some observations may also change with another parametrisation, e.g. via classifier optimisation (Gelbart et al., 2014) or parameter selection (Razavi et al., 2019), that could be more suitable for our type of data. In addition, a way to increase the detection window of an overtaking earlier in time could be to extend the database, which would also enable the use of data-hungry popular deep learning models such as Long Short-Term Memory (LSTM) networks (Zhang & Fu, 2020; Khairdoost et al., 2020) or Transformers (Guo et al., 2022). While we focused on traditional machine learning methods due to their robustness in low-data scenarios, we acknowledge that such advanced deep learning architectures may offer improved performance for time-series data like CAN signals. Contrary to the algorithms of this paper, such models exploit temporal dependencies in sequential data, capturing more intricate patterns in the signals, long-range dependencies and hierarchical temporal patterns, potentially providing a broader window and better performance. Recent advances in large language models (LLMs) open new possibili-

ties for multivariate time-series analysis, including vehicle sensor data, due to the LLM’s potential superior accuracy compared to traditional machine learning and even deep-based models such as CNNs and LSTMs (Razavi & Jamali, 2024). Although LLMs are primarily designed for textual data, their underlying architecture, the Transformer, has shown strong performance in time-series applications through variants like Informer (Zhou et al., 2021), Autoformer (Wu et al., 2021), and more recently, TST, PatchTST (Nie et al., 2023) and LLM4TS (Chang et al., 2025).

These models, however, typically require significantly larger volumes of labelled data to train effectively, which was a limiting factor in our current setup. In this future work direction, we are working on capturing large amounts of unlabeled data from the employed trucks by recording driving sessions continuously. This would allow to increase our dataset, for example, via pseudo-labelling (Li et al., 2019), selecting samples with high prediction probability as given by the classifiers trained with labelled data, thus enabling the use of the mentioned deep models. A larger dataset would also increase the diversity of signal patterns in non-overtake manoeuvres, identified as one of the issues emerging from the realistic and unconstrained nature of our data acquisition, where class diversity cannot be guaranteed a priori.

A unique part in future phases of the BIG FUN project will be the extension of the dataset with multimodal data, including video streams from digital cameras placed inside and around the truck, complementing traditional data such as CAN signals and Driver Interaction Input data. This will allow us to build a more holistic representation of truck usage and driver behaviour across different contexts. These multimodal data sources will be combined with AI-based analysis to automatically identify moments of potential significance in truck journeys, not only overtakes. Coupled with user experience (UX) studies and expert evaluation, these AI-discovered events are expected to generate actionable insights that contribute to safer, more intelligent vehicle systems and improved commercial mobility.

## **CRediT authorship contribution statement**

**Fernando Alonso-Fernandez:** Conceptualization, Writing – original draft, Writing – review & editing, Methodology, Software, Validation, Supervision, Funding acquisition. **Talha Hanif Butt:** Writing – original draft, Writing – review & editing, Data curation, Software. **Prayag Tiwari:** Conceptualization, Writing – review & editing, Methodology, Validation, Supervision.

## **Declaration of Competing Interest**

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## **Acknowledgements**

The authors thank the BIG FUN (Big Data-Powered End User Function Development) project of the Swedish Innovation Agency (VINNOVA) for funding this research (project 2021-05045). The authors would like to acknowledge Volvo Group for their collaboration and support in providing access to in-service truck data, which was instrumental in conducting this research.

## **Declaration of generative AI and AI-assisted technologies in the writing process**

During the preparation of this work, the authors used ChatGPT in order to proofread the text and improve its readability. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

## References

- Badue, C., Guidolini, R., Carneiro, P. V., Azevedo, P., Cardoso, V. B., Forechi, A., Jesus, L., Berriel, R., Paixao, T. M., Mutz, F. et al. (2021). Self-driving cars: A survey. *Expert Systems with Applications*, 165, 113816.
- Blaschke, C., Schmitt, J., & Färber, B. (2008). Predicting overtaking manoeuvres via can-bus data. *ATZ worldwide*, 110, 47–51.
- Breiman, L. (2001). Random forests. *Machine Learning*, 45, 5–32.
- Butt, T. H., Tiwari, P., & Alonso-Fernandez, F. (2024). Predicting overtakes in trucks using can data. In *Proc. 14th Scandinavian Conference on AI (SCAI)*. Jönköping, Sweden.
- Chang, C., Wang, W.-Y., Peng, W.-C., & Chen, T.-F. (2025). Llm4ts: Aligning pre-trained llms as data-efficient time-series forecasters. *ACM Trans. Intell. Syst. Technol.*, 16. URL: <https://doi.org/10.1145/3719207>. doi:10.1145/3719207.
- Chen, K., Liu, P., Li, Z., Wang, Y., & Lu, Y. (2021). Modeling anticipation and relaxation of lane changing behavior using deep learning. *Transportation Research Record*, 2675, 186–200.
- Das, A., Khan, M. N., & Ahmed, M. M. (2020). Detecting lane change maneuvers using shrp2 naturalistic driving data: A comparative study machine learning techniques. *Accident Analysis & Prevention*, 142, 105578.
- Das, A., Khan, M. N., & Ahmed, M. M. (2023). Deep learning approach for detecting lane change maneuvers using shrp2 naturalistic driving data. *Transportation Research Record*, 2677, 907–928.
- Dutra, I., & Fonseca, J. (2022). Data and knowledge for overtaking scenarios in autonomous driving. *Journal of Autonomous Vehicles and Systems OCTOBER*, 2, 041006–1.

- Fu, T., Li, Z., & Guo, L. (2021). Truck-related crash analysis and mitigation strategies: A comprehensive review. *Journal of Transportation Safety & Security*, 13, 496–520.
- Garcia, J., Serrano, J., Velez, J., & Campos, J. (2020). Can bus data analysis for the development of vehicle diagnostics and predictive maintenance systems: A review. In *2020 IEEE International Conference on Vehicular Electronics and Safety (ICVES)* (pp. 1–6). IEEE.
- Gelbart, M. A., Snoek, J., & Adams, R. P. (2014). Bayesian optimization with unknown constraints. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence UAI'14* (p. 250–259). Arlington, Virginia, USA: AUAI Press.
- Guo, H., Keyvan-Ekbatani, M., & Xie, K. (2022). Lane change detection and prediction using real-world connected vehicle data. *Transportation Research Part C: Emerging Technologies*, 142, 103785.
- Hasan, M., Bashir, S., & Jalal, A. (2021). Multi-sensor data fusion for intelligent driver assistance systems. *Sensors*, 21, 3810.
- Haykin, S. (2009). *Neural Networks and Learning Machines*. Pearson Education.
- Huang, X., McGill, S. G., Williams, B. C., Fletcher, L., & Rosman, G. (2019). Uncertainty-aware driver trajectory prediction at urban intersections. In *2019 International Conference on Robotics and Automation (ICRA)* (pp. 9718–9724).
- Khairdoost, N., Shirpour, M., Bauer, M. A., & Beauchemin, S. S. (2020). Real-time driver maneuver prediction using lstm. *IEEE Transactions on Intelligent Vehicles*, 5, 714–724.
- Kim, I.-H., Bong, J.-H., Park, J., & Park, S. (2017). Prediction of driver's intention of lane change by augmenting sensor information using machine learning techniques. *Sensors*, 17, 1350.

- Kumar, A., Saini, A., & Singh, D. (2020). Recent trends in adas technologies: A review. *Journal of Intelligent Systems*, 29, 899–909.
- Li, Z., Ko, B., & Choi, H.-J. (2019). Naive semi-supervised deep learning using pseudo-label. *Peer-to-Peer Networking and Applications*, 12, 1358–1368.
- Lin, H.-Y., Dai, J.-M., Wu, L.-T., & Chen, L.-Q. (2020). A vision-based driver assistance system with forward collision and overtaking detection. *Sensors*, 20.
- Lin, Y.-C., Lin, C.-L., Huang, S.-T., & Kuo, C.-H. (2021). Implementation of an autonomous overtaking system based on time to lane crossing estimation and model predictive control. *Electronics*, 10, 2293.
- Liu, P., Qu, T., Gao, H., & Gong, X. (2023). Driving intention recognition of surrounding vehicles based on a time-sequenced weights hidden markov model for autonomous driving. *Sensors*, 23.
- McKnight, C. E., & Bahouth, G. T. (2020). Factors influencing truck accident severity on major highways. *Transportation Research Record*, 2674, 607–616.
- Mozaffari, S., Hammad, A. H., & Ponnambalam, L. C. (2020). Deep learning-based models for overtaking behavior prediction in autonomous vehicles. *Transportation Research Part C: Emerging Technologies*, 118, 102686.
- Nalcakan, Y., & Bastanlar, Y. (2023). Lane change detection with an ensemble of image-based and video-based deep learning models. In *2023 Innovations in Intelligent Systems and Applications Conference (ASYU)* (pp. 1–7).
- Neumeier, S., Hildebrandt, A., & Duran, J. M. (2019). Impact of can bus data quality on vehicle predictive maintenance systems. *Automotive Innovation*, 2, 312–321.
- Nie, Y., H. Nguyen, N., Sinthong, P., & Kalagnanam, J. (2023). A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*.

- Ortega, J., Ortega, M., Ismael, K., Ortega, J., & Moslem, S. (2024). Systematic review of overtaking maneuvers with autonomous vehicles. *Transportation Engineering*, 17, 100264. URL: <https://www.sciencedirect.com/science/article/pii/S2666691X24000393>. doi:<https://doi.org/10.1016/j.treng.2024.100264>.
- Pourjafari, N., Ghafari, A., & Ghaffari, A. (2024). Navigating unsignalized intersections: A predictive approach for safe and cautious autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 9, 269–278.
- Razavi, H., & Jamali, M. R. (2024). Large language models (llm) for estimating the cost of cyber-attacks. In *2024 11th International Symposium on Telecommunications (IST)* (pp. 403–409). doi:[10.1109/IST64061.2024.10843617](https://doi.org/10.1109/IST64061.2024.10843617).
- Razavi, H., Sarabadani, H., Karimisefat, A., & LEBRATY, J.-F. (2019). Profitability prediction for atm transactions using artificial neural networks: A data-driven analysis. In *2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI)* (pp. 661–665). doi:[10.1109/KBEI.2019.8735037](https://doi.org/10.1109/KBEI.2019.8735037).
- Scheel, O., Nagaraja, N. S., Schwarz, L., Navab, N., & Tombari, F. (2019). Attention-based lane change prediction. In *2019 International Conference on Robotics and Automation (ICRA)* (pp. 8655–8661).
- Stefansson, E., Jiang, F. J., Nekouei, E., Nilsson, H., & Johansson, K. H. (2020). Modeling the decision-making in human driver overtaking. *IFAC-PapersOnLine*, 53, 15338–15345.
- Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer-Verlag New York, Inc.
- Wang, W., Qie, T., Yang, C., Liu, W., Xiang, C., & Huang, K. (2022). An intelligent lane-changing behavior prediction and decision-making strategy

- for an autonomous vehicle. *IEEE Transactions on Industrial Electronics*, 69, 2927–2937.
- Wu, H., Xu, J., Wang, J., & Long, M. (2021). Autoformer: Decomposition transformers with Auto-Correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems*.
- Xie, D.-F., Fang, Z.-Z., Jia, B., & He, Z. (2019). A data-driven lane-changing model based on deep learning. *Transportation Research Part C: Emerging Technologies*, 106, 41–60.
- Xing, Y., Lv, C., Wang, H., Cao, D., & Velenis, E. (2020). An ensemble deep learning approach for driver lane change intention inference. *Transportation Research Part C: Emerging Technologies*, 115, 102615.
- Xing, Y., Lv, C., Wang, H., Wang, H., Ai, Y., Cao, D., Velenis, E., & Wang, F.-Y. (2019). Driver lane change intention inference for intelligent vehicles: Framework, survey, and challenges. *IEEE Transactions on Vehicular Technology*, 68, 4377–4390.
- Xu, T., Zhang, Z., Wu, X., Qi, L., & Han, Y. (2021). Recognition of lane-changing behaviour with machine learning methods at freeway off-ramps. *Physica A: Statistical Mechanics and its Applications*, 567, 125691.
- Zhang, H., & Fu, R. (2020). A hybrid approach for turning intention prediction based on time series forecasting and deep learning. *Sensors*, 20, 4887.
- Zhang, Z., Zhao, B., & Zhang, T. (2021). Driver intention recognition using deep learning techniques: A review. *Sensors*, 21, 4240.
- Zheng, Y., & Hansen, J. H. (2017). Lane-change detection from steering signal using spectral segmentation and learning-based classification. *IEEE Transactions on Intelligent Vehicles*, 2, 14–24.
- Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2021). Informer: Beyond efficient transformer for long sequence time-series

forecasting. In *The Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Virtual Conference* (pp. 11106–11115). AAAI Press volume 35.