Technical University Delft

Master thesis

Predicting the user's next action on the in-car infotainment system

J.T. Haringsma
4735757

Report submitted to:
Joost de Winter
Company supervisor: Marco Wiedner
Additional supervisor: Enrico Mion

June 9, 2022

Preface

After using the pure technical expertise gained during my courses in Delft, I was missing the involvement of humans in my work. So when Porsche offered me the opportunity to study the behaviour of their drivers while using my technical expertise, I had no doubt and gladly accepted this offer. Little did I know beforehand what experience and lessons this would bring me. While working on my thesis in a different city and even a different country, I'm very glad with the results I achieved and can present in this report.

However, this was not possible without the help of others. First, I would like to thank Porsche and especially the EPD-department for enabling me to conduct my research and offer me the resources necessary for this. Especially I would like to thank Marco Wiedner, who guided me in this whole process and played a major role in creating the work environment which we were allowed to enjoy. Marco, thank you for your support and the Swabian language lessons.

Additionally I would like to thank the wonderful people I met in Weissach, who were there to have a laugh, but also to provide guideness in my process. So, thank you Francesco, Thomas, Satiyabooshan and Enrico.

Thanks to the Technische Universiteit Delft and in particular Joost. Thank you for all the meetings we had and the advice you gave. I learned not only a lot about research but also how to deal with a company and their policies, which was a very valuable lesson. Thank you for constantly seeing the bigger picture, which was sometimes a bit blurred for me.

Lastly, I would love to thank my family for supporting me throughout this long period of studying: after this, it is really finished, I promise! And Sanne, I'm glad you were part of this process. Thank you for all your support, which was more than you probably realised.

I hope you enjoy reading my work!

Jelmer

1 Abstract

The main task during operating an automotive vehicle is driving. Nowadays, distractions form a potential risk of claiming the workload necessary for the driving task. Interacting with the User Interface (UI) of the vehicle can be such a distraction. Predicting the next action on the UI can help decrease the risk of distractions. To predict the next user action, knowledge about the UI interaction behavior is necessary.

In this work, a descriptive analysis is conducted on a large naturlistic dataset which has not been matched in size by other related work in the field of automotive Human Machine Interface (HMI) research. The analysis is conducted to gain insight in the interaction behavior of the drivers. This behavior is related to several driving task conditions such as occupancy, speed and drive mode.

The results of the descriptive analysis are used to implement several prediction models to predict the next user action. Among the models, the Long Short-Term Memory (LSTM)-network achieved a validation accuracy of 87%. This work shows the potential for analysing UI interaction behavior and leveraging this for prediction purposes. It could help in forming a foundation for future adaptive UI work.

Contents

List of Acronyms

2 Introduction

Distractions during driving are responsible for 8.7% of the total amount of fatal accidents in 2019 [1]. After 2019 this increased to almost 10%. Distractions or secondary tasks may involve three different types of demands: visual demands by taking your eyes off the road; manual demands by taking your hands off the wheel and thirdly, cognitive demands by taking your mind off the task of driving. As people are getting more connected, this connectivity is also taken to the road and in the cars. The infotainment system or the mobile phone can require this demand and form a distraction [2][3].

To minimise the use of phones and the infotainment system and their risks for distractions, several solutions were designed. An example of a tailored solution for the automotive market is a system limiting the options of the mobile phone given a workload condition [4]. This system blocks calls or mutes notifications when a certain workload condition is detected by the steering wheel angles and driving speed. However, limiting phone functions may not be attractive for drivers who are dependent on the GPS connection or music player while on the road [5]. To reduce the manual demands, hands-free systems are increasing in popularity and Fitch at al. [6] demonstrated that such systems improved traffic safety over hands-on systems. Nonetheless, the National Safety Council [2] claimed that eliminating all types of mobile phone usage will always be the safest option. Hands-free calling decreases the visual and manual demands, but it still causes cognitive distraction.

Other solutions can be found in the field of real-time safety warnings where the driver is alerted when a distraction is recognised. This real-time safety alert can be performed by gaze analysis, but this is found to be intrusive and not always accepted by drivers [7]. So, Chai et al. [8] proposed a method which tracks lanes and reports when there are any deviations from this lane. This lane keeping method is non-intrusive and is able to detect distractions during driving. Contrary to the method of Chai et al., which focused on the effect of distractions, eliminating the root cause could make this effect-method obsolete. One way to address this is by improving the accessibility of the infotainment system to decrease the overall distraction time.

The UI is the point of contact to control the software through the hardware. The type of UI includes: command-line interface, text interface, voice interface and graphical interface [9]. The latter is the main focus of this study. Through this interface, the user can control a wide range of functions in the car. According to Falk, the graphical UI has the following options: it can either be static (no change over time), manually adaptable (the position and application can be adjusted manually by the user), or adaptive. An adaptive UI can change automatically depending on the needs of the driver based on the driving task condition and/or the type of application. In the current study, the needs of the driver and the driving task conditions will be analysed to increase knowledge of interaction behavior.

To understand the interaction behavior, knowledge should be obtained on how the driving task conditions affect driver UI inputs. This expertise of in-car HMI usage can be associated with three conditions: driving speed, drive mode and passenger presence. Senders showed that the visual sampling frequency and therefore the cognitive workload, increases at high driving speed [10]. Thus, at higher driving speed, cognitive workload increases. This is why higher speeds are more demanding and less attention can be given to secondary tasks (e.g. interaction with the UI). In addition, Melman et al. showed that the driving behavior of the participants alters when the drive mode changes [11]. As an example, the driving task is perceived to be more sporty when switched to sport mode which results in increased acceleration initiated by the driver. The focus necessary for the driving task is impacted by this perception and influences the interaction behavior. Also external factors, such as the presence of a co-driver, can influence focus of the driver. Waugh explained that the presence of a co-driver can be labeled as a distraction comparable to using a mobile phone [12]. The occurrence of the aforementioned conditions, affects the way the UI interaction is performed. Bah et al. showed that the interaction and driving behavior of drivers in different scenarios affects the input method such as tactile, touch, and gesture. [13].

An in-depth descriptive analysis of numerous driving situations is conducted to validate if driving speed, drive mode and passenger presence affects the interaction behavior on the car's UI. This analysis is conducted based on the following research question: **Can the next user action be predicted based on the interaction history and specific driving task conditions?** Several sub-questions are formulated to support the main research questions. The sub-questions are based on the findings in literature and therefore split up as shown in Table 1. To answer the questions and make these insights available, a large naturalistic driving study is conducted by Dr. Ing. hc F. Porsche AG (Porsche) and used for the analysis. The purpose of this report is twofold: describing the UI interactions based on specific driving task conditions and subsequently use these findings to train a model to predict the next user action on the UI. The final model predicts the next action performed by the (co)driver to form the foundation of the eventual adaptation of the UI.

Chapter 3 shows the related work in the field of in-car HMI and methods to predict the next action. Chapter 4 discloses all the properties of the earlier described data set and Chapter 5 describes the analysis performed on this data. Chapter 6 combines the aforementioned sections into a prediction model which predicts the next user action on the UI.

Research question	**Important feature**
What interaction characteristics on the UI in terms of interactions/hour, total number of interactions or interaction time can be observed when the user is driving alone or with a co-driver?	Occupancy
What interaction characteristics on the UI in terms of interactions/hour, total number of interactions or interaction time can be observed as a function of speed?	Speed
What interaction characteristics on the UI in terms of interactions/hour, total number of interactions or interaction time can be observed during different drive modes?	Drive mode

Table 1: Research questions on the effect of specific driving task conditions on UI interaction behavior

Contribution

The goal of this study is to predict the next user action for future adaptive UIs. To predict the next user action, a good understanding of the important variables which influence the behavior is necessary. An extensive naturalistic driving study is done to provide access to the interaction behavior and the influence of external variables. A descriptive analysis has been performed on this dataset to determine the variables which has an effect on the interaction behavior. These variables are used as input for several next user action prediction models. State Of The Art (SOTA) models will be used and performance of the models on the dataset will be compared. The contribution of this work includes:

- Conduct a naturalistic driving study collecting a large dataset.
- Filter and preprocess the dataset for further analysis.
- Analyze the preprocessed dataset to find the relations between external variables (e.g. passenger presence, speed, drive mode) on the interaction behavior.
- Predict the next user action using external variables (e.g. passenger presence, speed, drive mode) with SOTA models.

An additional contribution of this work is presenting a large scale study (7500 recorded driving hours) in the field of automotive HMI which shows the potential and possibilities to leverage naturalistic interaction data and can help provide insight in the interaction behavior of the driver and passenger. As a result, multiple areas of the industry can take advantage of this work, like UI concept and design, software development, driving assistance, safety departments and psychologists.

3 Related Work

Analysing user interactions on the HMI within the automotive domain was previously done by Wolf et al. [14]. In this work, the input method was predicted based on its historical usage and the driving context. This pilot study was based on 11 anonymous driving sessions with a maximum of 90 minutes each. Three different input methods were used in this dataset: touch, hard keys and speech input. Based on the performance comparison of several methods, a decision tree was used to predict the input method. It was recommended not to use Deep Neural Networks (DNN) because of the small feature space, which increases the risk of over fitting. Wolf et al. achieved a mean accuracy of 77.2% in the 10-fold cross-validation. Other related work on user action prediction in the automotive industry is non-existent. Inspiration can be taken from the field of Human Computer Interface (HCI) and time series prediction work (e.g. stock market forecasting and anomaly detection).

In the HCI domain, Song et al. introduced the MAST-model for predicting and analysing user behavioral patterns using sensors of a mobile phone [15]. The MAST-model is a probabilistic model which predicts $P_{\Delta_t}(Y|X)$: the probability of the actions, Y_i, happening at time $t + \Delta t$, given the observation of the previous action (X(i)) detected at time t. The predicted action was based on the highest probability determined by the model and the historical input. One presented use-case is the *Proactive UI Adaptation*, an automatic UI adaptation. This adaptation is based on the behavioral patterns of the user, the time of the day and movements performed by the user. With the prediction results, the UI can be adapted so that the predicted action appears in a more prominent location on the front screen. The transformation of this concept to a working framework has not been done yet.

Sarker took a different approach in relating phone actions such as taking, rejecting or missing phone calls to contextual variables [16]. The used dataset consisted of records where the action is linked to the corresponding context, for instance rejecting a phone call from your mother in the office at 12 am. This timestamp of the observation is recorded to transform the data to time series. Prediction is done by training a decision tree algorithm on data that has been cleaned from noise with a Naive Bayes classifier. The decision tree algorithm outperformed the base models. Suggestions were made to pursue this work further in the field of recommendation systems. Connecting user contextual information to user preference is proven to be effective for increasing prediction accuracy by Wolf et al. and Lee et al. [14] [17]. A decision tree filters the data based on characteristics of the historical data to form a road map for the prediction. Random Forest (RF) expands this by combining multiple decision trees into a forest. Wolf et al. used RF to predict the input method. Karashu, Moore and Hengl also used RF for predicting next time steps [18][19][20]. RF outperformed the stated statistical baseline models by getting the best accuracy, F1-score, precision and recall.

Lee et al. proposed spatiotemporal structure learning, where a Bayesian classifier was used to adapt a UI on a mobile phone [17]. Several contextual variables were used; location, activities, weather, time and user-labeled emotion. Based on these variables, the Bayesian model was able to achieve a top-3 accuracy of 69%. This again showed the effectiveness of connecting the spatiotemporal relations within the data.

Since general time series prediction problems are widely applicable, inspiration can be taken from these models for implementing SOTA models from the field of standard time series prediction in user prediction use cases. These models use the same structured dataset, which makes them applicable for the user action prediction usecase.

One of the methods is Support Vector Machine (SVM), which is a linear model for classification and regression problems. A line or plane is fit to the data to separate it into different possible classes, where the classes can be next user actions or more general: a value at the next time step. Sapenkevych et al. showed that SVM is a promising method to use on time series prediction use cases [21], like financial market prediction and environmental parameter forecasting. Both usecases are dependent on a 1D historical dataset. For a higher dimensional dataset, a kernel function must be used to transform the non linear high dimensional input space to a linear 1D space. This makes SVM suitable for being used on (multidimensional) spatiotemporal datasets for user action prediction.

AutoRegressive Integrated Moving Average (ARIMA) has been known since 1970 as the standard method for time series prediction [22]. As the name suggests, the ARIMA uses a lagged moving average filter to make the time series smoother, and the auto-regressive part assumes implicitly that the future has already been seen in history. As the model has been around since 1970 and the demand for more complex datasets becomes larger, the ARIMA model might find some difficulties, mainly in finding deeper correlation within the data through the risk of overfitting on a single variable [23]. However, the historical performance in stock price forecasting and infection rate prediction makes the model interesting for complex use cases [24] [25].

Other work involved more complex models in the field of deep learning (e.g. Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN)), coming from the fields of image and speech recognition [26][27]. Selvin et al. used CNN and RNN on time series [28]. These authors showed that deep models are capable of making predictions based on hidden relations in the historical data, which were less likely to be found by statistical models. This is demonstrated by Siami-Niami et al. where LSTM is making better predictions than ARIMA in terms of accuracy [29].

CNN outperformed the RNN (LSTM) in the work of Selvin et al. by showing a better fit on the predicted stock price than the LSTM [28]. However, these predictions were made on a 1D stock price dataset, which does not necessarily imply this performance on a multi-dimensional data. Also, Li showed that LSTM - with some tuning - could beat the baseline models in terms of accuracy meteorological, indoor temperature and human movements prediction [30].

The spatiotemporal relations within the dataset can be brought to the surface by having a profound understanding of the data. This understanding can be achieved by performing a descriptive data analysis. The following chapters will show an explanation of the dataset and the descriptive analysis. The results of the descriptive analysis will be used to form the features. These features are used as input to compare several baseline models which includes the LSTM, Random Forest, VARMAX and SVM based on the results mentioned before.

4 Used dataset and preprocessing

4.1 Dataset

To collect the data used for the continuation of this research, an anonymised data collection campaign was initiated to create a large-scale naturalistic dataset containing a total of 1836 cars, 1500 of which are Porsche 911s (992 generation) and 386 Porsche Panamera (G2 generation). This large scale, 7500+ hours of driving, is due to the number of participants; all Porsche customers who complied with the privacy and product improvement statement were participating in this study. The vehicles are equipped with an over-the-air transmission module for CAN-BUS and HMI data, which transmits the data with a frequency of two minutes to the backend. The signals are received in time series format with a timestamp connected to the changed state. The GPS information is not collected, just as the Vehicle Identification Number (VIN) and any personal characteristics of the driver, resulting in a completely anonymised dataset. The campaign is collecting from March 14, 2021, until January 31, 2022. See Figure 2 for the collection time period. The collected signals are bundled into sessions, where every drive is recorded as a separate session, from starting until shutting off the car. The session length is limited to 85 minutes and continues into a new session when the previous session exceeds this limit.

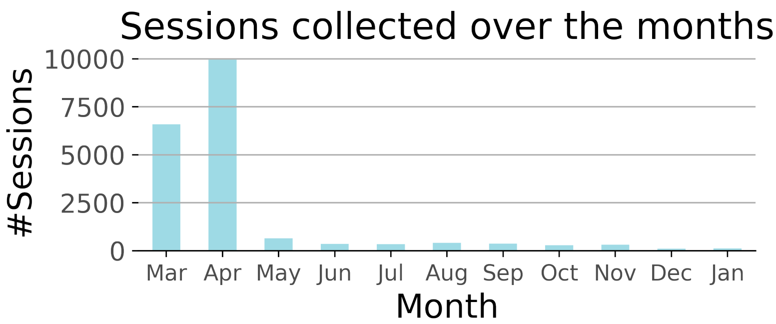


Figure 2: Collected sessions from March 2021 until January 2022

Figure 3 shows the average length of the collected sessions. The high peak at minute 85 can be described as the point where the session is stopped after 85 minutes before it starts a new session. In total 47,010 sessions were recorded, with 33,117 session coming from the Porsche 911 and 13,893 session from the Porsche Panamera.

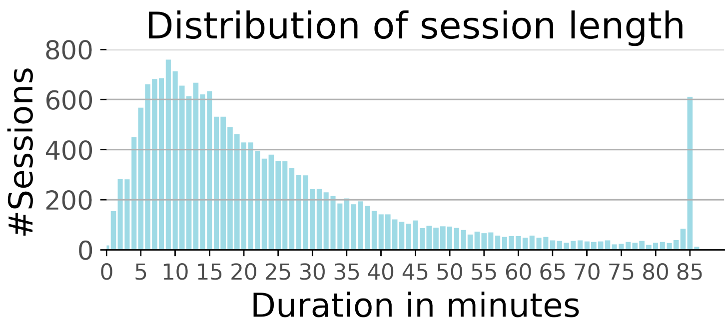


Figure 3: Number of session recorded with a specific length

(a) Complete overview of infotainment system, including instrument cluster, center screen, steering wheel buttons etc

(b) Center screen of the infotainment system

Figure 4: Infotainment system of the recorded vehicles

As mentioned in the introduction, speed and occupancy played a crucial role in the UI interaction behavior of the drivers. To investigate the effect of these variables on the interaction behavior, two CAN/BUS signals were collected: *AB_Gurtschloss_Reihe2_BF* and *KBI_angez_Geschw*, which indicate respectively the presence of a passenger and the current velocity of the car. Next to the occupancy and speed signals, all the HMI signals coming from the infotainment system were collected. These signals indicate which actions were taken on the infotainment system. Such actions include touches on the center screen (softkeys), keys on the mid console (hardkeys), voice commands and steering wheel buttons (see Figure 4). Besides direct button presses, higher-level input recognised by the HMI's software, such as confirmation of a started route, were recorded. Other possible recorded values are shown in Table 3.

To specify the HMI signals, each signal can be linked to a certain domain. A domain is a section of the current infotainment system and represents a cluster of functionalities. For example, the car domain consists of all the settings related to the car (e.g. chassis-, exhaust- and engine-settings). The navigation domain covers all the attributes related to route guidance (e.g. map view, destination input, route guidance details etc). A description of the domains is shown in Table 2. The menu of the UI is designed in a way that every screen is part of a specific domain. The location in the menu can therefore be explained by the specific screen or the domain. The following domains, as listed in Table 2, can be observed in the data.

Domain	Description
Car	Settings related to car setup
Navigation	Navigation settings and route guidance
Tuner	Radio services and settings (FM/DAB+/AM)
My Screen	Manually customisable menu of selected icons
Media	Media services for all playback devices
Telephone	Telephone services for phone calls
Terminal Mode *	Mode when shutting down the car
Connectivity *	Service for establishing new connections (Bluetooth, phones)
Address book *	Access to list of contacts for calls/mails
Online *	Car-UI internet browser
Tone *	Adjust the music tones (bass, balance, etc.)
Settings *	General settings for the UI
User Admin *	Administrator settings (Porsche ID)
Update *	Updates regarding the operating system
Messaging *	Service for messaging (SMS etc.)
Boardbook *	Searching through the infotainment system
Ecall *	Calling the emergency service

Table 2: Domains and Description of all available domains in the Porsche UI (*combined into domain *Other*)

The total time spent in each domain is summed to calculate the percentage of usage of that specific domain. Figure 5 shows this breakdown of domain usage over total time. All domains that were used less than 5% of the total time are marked with a * in Table 2 and are combined in *Other* to keep it clear.

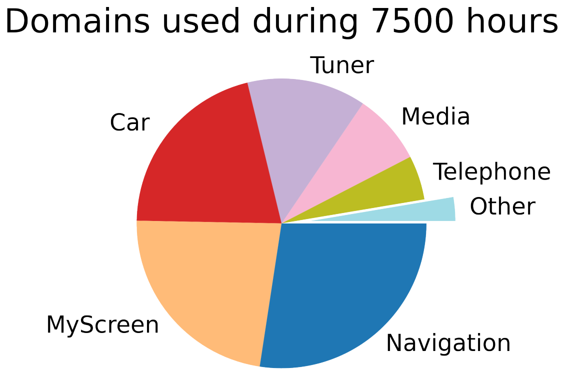


Figure 5: Domain usage over total time

4.2 Data validation

After an inspection of the customer dataset, anomalies were observed in the data. These anomalies included unexpected and unknown values which were not generated by the data collector before. A validation campaign was conducted to verify if the dataset was not bugged. A Porsche 911, generation 992, was used and made ready to collect the data with the infotainment system data-collector. The validation campaign consisted out of one car, but was further exactly identical to the customer campaign. To validate the signals and the corresponding payload, video footage of interactions performed on the UI in combination with the signals from the infotainment system data-collector (validation campaign) were collected. A GoPro camera was mounted to the driver's seat to record the complete cockpit and the timestamp of the video was matched to the car's time for synchronisation purposes. The validation started by interacting with the UI while recording it with the infotainment system data-collector and camera. Afterwards, the timestamps of the dataset were matched with the timestamps of the video. The signals coming from the interactions recorded by the data collector were compared with the interactions visible on the video recording.

After inspection, the payload coming from the phone signal was marked as unstable and is therefore skipped in the dataset. The other payloads of the signals were found valid and therefore kept and used in the continuation of the analysis.

4.3 Preprocessing

The goal of the preprocessing was to deliver a ready-to-use dataset with all the features necessary to analyse the effect of every variable on the interaction behavior. The first step of the preprocessing was to store the data in a place that allows processing. Since the data is stored on the Porsche Data Platform (PDP), which does not offer the ability to process large amounts of data, a processing pipeline was built. Since the data collector is collecting all the signals, HMI and CAN/BUS signals combined, loading all the data through the pipeline would result in exceeding the memory limit of the instance and connection. Therefore, data pre-filtering was done during accessing PDP using SQL. As described in Chapter 4.1, the information of the seatbelt, speed and infotainment were selected. A secured connection was made between the PDP and Amazon Web Services (AWS), where the pre-filtered data was stored in an AWS S3 bucket. AWS was used to provide processing power and flexible storing capacity. AWS provides instances that are variable in the amount of memory. For this purpose 'ML.r5dn.8xlarge' with 256GB of RAM was used on an AWS Sagemaker setup. To execute the data processing the Python programming language is used (Python Software Foundation, https://www.python.org/). Pandas package was used to analyse and visualise the data. Since the data collector samples every signal with a timestamp, but sends the signals in unsorted batches to the cloud, data was sorted before any other filtering was done. Some errors were observed in the registration of the Unix timestamps connected to signals, timestamps from January 1, 1970 (Unix timestamp is 0) were observed in the data. Twenty-four sessions containing timestamps not corresponding to the remaining of the session were removed. Furthermore, sessions where the car was not moving for more than 30 minutes were removed. There were in total 610 sessions, where this 30+ minute break was noticed. Another 448 sessions were removed which had an average speed of 0 km/h. In total 28,010 sessions contained zero interactions were removed, since these would not contribute to finding the relation between the context and the interactions. The filtering process is shown in Figure 6, where the yellow block represents the initial dataset, the red block the deleted/filtered sessions, and the green shows the results of the filter. The grey arrows show the filtering criteria, which causes the initial data to divide into the next red and green blocks.

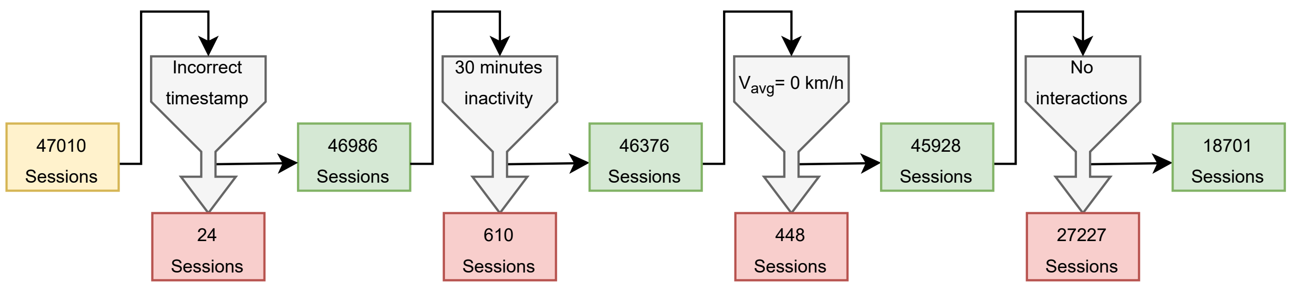


Figure 6: Filtering process of the data where anomalies (in red) are deleted

The interactions were sampled on-change, while the speed (when driving) was sampled with a frequency of 20 Hz. The speed signal was therefore used as a timeline for the other signals; the on change context-signals were added at their occurrence. The context signals were forward filled into a separate column. This way, the following features were created: Current domain, Occupancy, Speed, Media, Navigation, and Input method.

The values of these features were automatically extracted from the data collector. Table 3 shows all the possible values. The different features followed from the research questions in Table 1. With these features added, the actions like soft-touch, hardkey and speech can be separated from the non-action data points. This results in a dataset of only actions with the corresponding context, which forms the final dataset used in the analysis. The final (preprocessed) dataset contains 18,781 sessions, coming from 13,021 sessions with a 911 and 5,760 sessions from a Panamera.

Features	**Possible values**
Current domain	Navigation, MyScreen, Car, Tuner, Media, Telephone, Other
Occupancy	1 occupant, 2+ occupants
Speed	1,2,3 .... 320,321, 322
Media source	FM, Bluetooth, USB, AM, DAB+, Phone
Route guidance	Active, non active
Input Method	Soft touch, hard key, speech
Drive mode	Normal, Sport, Sport Plus, Wet, Individual, E-Charge*, E-Hold*, E-Power*, Hybrid*

Table 3: Possible values for every added feature and driving task condition

5 Exploratory data analysis

5.1 Introduction

The relation between the introduced variables and the interaction behavior, as shown in Chapter 2, is still unknown. To find the influence of specific driving task conditions on the actual interaction behavior of the user on the UI, an analysis is performed. Several studies showed the significance of one single variable (Chapter 2). The dataset as described in Chapter 4, offers multiple variables to analyse the combined effect on the interaction behavior.

5.2 Method

The dataset is (pre-)processed as described in Chapter 4.1 to make it suitable for further analysis. This analysis explores the impact of different variables on the interaction behavior of the user. Three normalisation options are considered to compare these variables: normalisation over time per feature, normalisation to interactions per hour, or normalisation to the number of interactions.

To normalise over time, the duration of a certain feature was calculated by summing all the time of a certain feature and divide by the summed total time. To calculate the interactions per hour, the interactions during a certain feature were counted and divided by the total time spent at this feature. The last option, normalising over interactions, was calculated by the number of interactions done during a feature compared to the total number of interactions.

As Table 3 shows, the possible values for the speed range are from 1 to 322 with a resolution of 1 km/h. To decrease complexity, the speed is redefined by equation 1, which created 11 speed classes. After experimenting with different resolutions, a higher resolution would result in a cluttered set and lower resolution would not decrease the complexity enough. The 11 speed-groups still represent the actual distribution of the speed well. Figure 7 shows the average speed distribution of the sessions. The low-speed groups represent all the idling at rest and acceleration from a standstill when the speed is between 0 and 9 kilometers per hour. The faster groups are defined by blocks of 20 km/h until 209 km/h.

$$v_{rounded} = \left\lfloor \frac{v_{actual} + 10}{20} \right\rfloor \cdot 20 \tag{1}$$

Figure 8 shows the amount of driving time. The faster speed blocks (+209 km/h) is marginal in terms of the total time. COnsequently, the analysis was stopped at 209 km/h. The height of the low-speed block in Figure 8 visualises that 15% of the time the car drives between 0 and 9 km/h.

5.3 Results

In total, there were 4577 recorded driving hours with one occupant, of which 3103 hours from the 992 and 1474 from the Panamera. In total there were 2813 hours recorded with two occupants of which 1947 from the 992 and 865 from the Panamera. To compare these two groups, they are normalised to interactions per hour. The average trip length of drives with one occupant is 22 minutes and the trip length with two occupants is on average 31 minutes.

Figure 9a shows the relation between the number of occupants and the number of interactions per hour. A distinction has been made between the different car types present in the dataset.

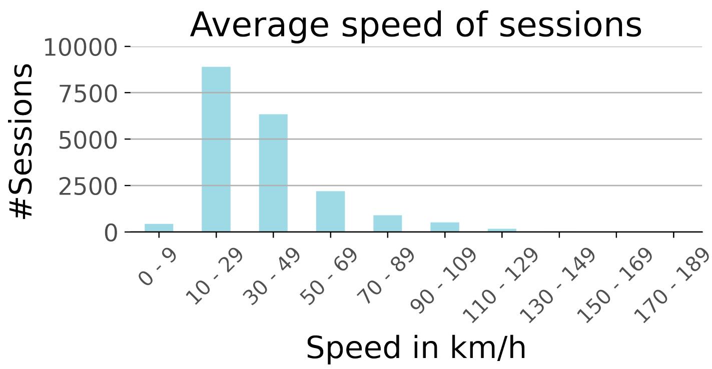


Figure 7: Average speed of sessions

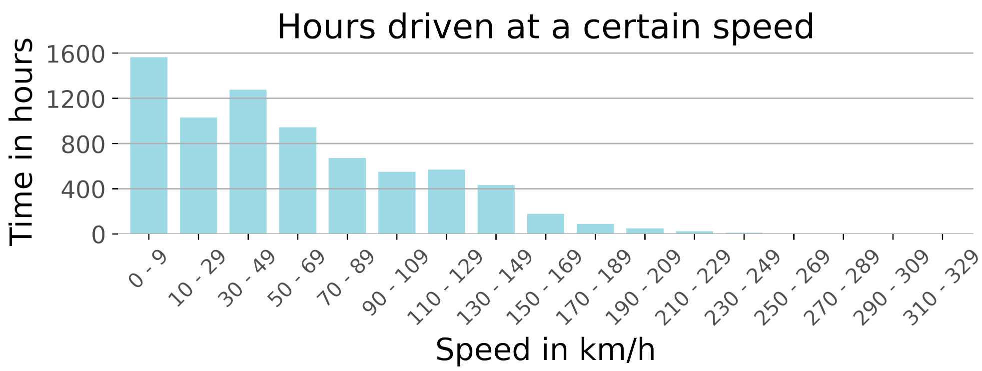
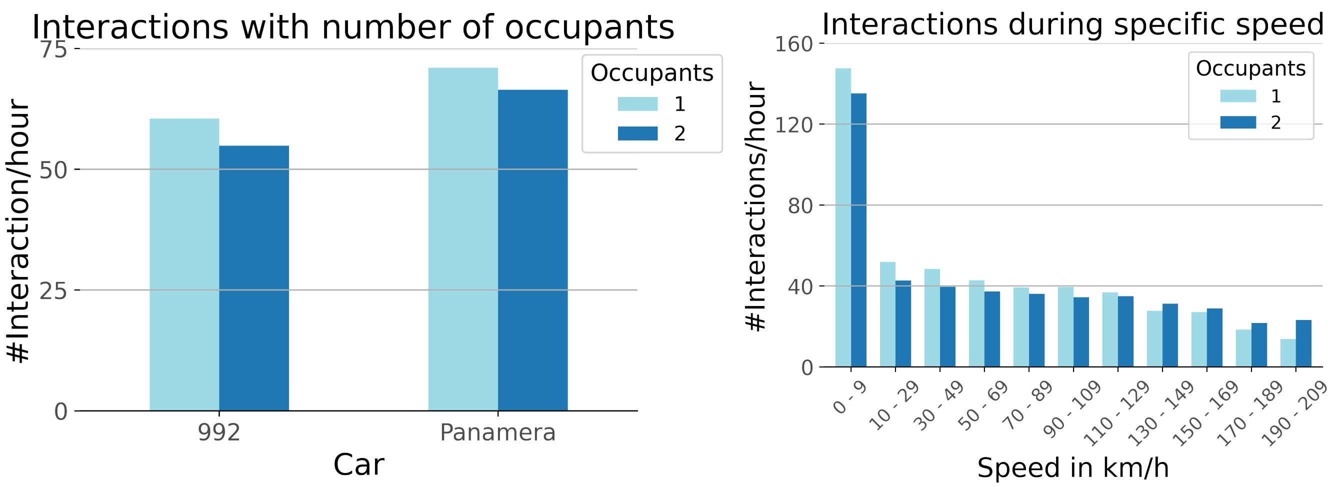


Figure 8: Time spend driving within speed block



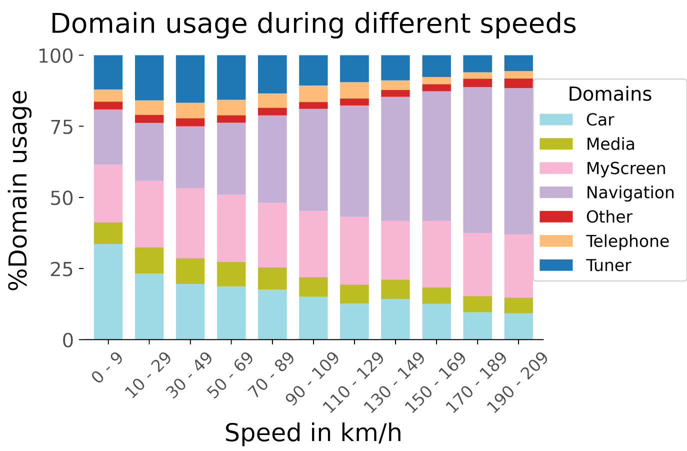
(a) Interactions per hour grouped by car and occupancy

(b) Interactions per hour grouped by occupancy per speed block

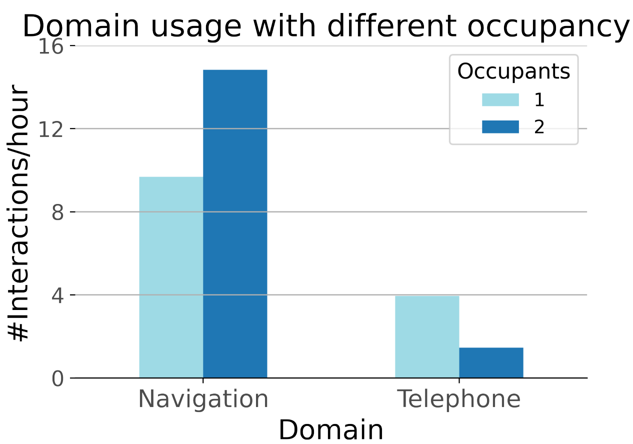
Figure 9: Occupancy analysis

To examine whether the speed impacts the infotainment usage, the interactions while driving at a certain speed are compared in Figure 9b. The interactions are normalised by interactions per hour, to correct for the difference in amount of available data for both features. An overall negative correlation was observed between speed and interactions per hour. A One-way-ANOVA test showed that the interactions of at least two speed group were significant, $F(10, 1690) = 100.23$, $p < 0.001$. To validate if the occupancy in the car makes an impact on the infotainment usage, two groups were distinguished: drives with one occupant and drives with two occupants. When the speed increases (speed >120 km/h), drives with two occupants show more interactions per hour than drives with one occupant. At lower speeds, drives with one occupant show more interactions per hour in comparison to two occupants. A T-test showed the significant difference of the interaction behavior between drives with two occupants and one occupant, t(9350) = 100.23, $p < 0.001$.

To add information to a specific interaction, the interactions were expressed in the domain they were performed in. Figure 10 shows the correlations between domain usage over the speed. Figure 10a shows the combined usage of the domains used during drives with two occupants and one occupant. The data is normalised over the time in the interest of comparison, because of the negative correlation between speed and number of interactions per hour as shown in Figure 9b. An relative increased usage of the navigation domain can be observed, where at lower speeds (<30 km/h) only 20% of time the navigation domain is used, while at higher speeds (>140 km/h) more than 50% of the time the navigation domain is used. The car domain decreased from 33% at lower speeds to less than 10% at higher speeds. The remaining domains stayed constant while the speed increased. A one-way-ANOVA was performed on the overall interactions in different domains along the speed range and showed a significant effect: F(76, 165) = -3.445, $p < 0.001$. Figure 10b shows the interactions in two different domains per occupancy, which were significantly different (navigation: (r(9350) = 2.3, $p < 0.001$ and phone: r(9350) = 1.7, $p < 0.001$). In the navigation domain, there were 1122 recorded hours for one occupant and 847 hours for two occupants. In the phone domain, there were 296 recorded hours from one occupant and 58 hours from two occupants, in total 355 hours of phone domain usage.



(a) Normalised percentage domain usage over time grouped per speed block.



(b) Mean number of interactions per hour grouped by selected domains and occupancy.

Figure 10: Domain usage over speed

Drives with two occupants show that the navigation domain is used 15% of the time, while drives with one occupant show that only 10% of the time the navigation domain is used. Other differences can be observed in the phone domain. During drives with one occupant, the phone domain is used 4% of the total time in comparison to drives with two occupants where only 1% of the time the phone is used.

To investigate the driving behavior, the drive mode plays an important role [11]. The drive modes available in the Porsche vehicles are indicated in Table 5. The interactions during different drive modes are shown in Figure 11a. Table 4 shows the usage of the drive modes per occupancy in hours.

Drive mode	**1 Occupant**	**2 Occupants**
Normal	3792	2229
Sport	407	301
Sport Plus	59	82
Wet	58	32
Individual	66	32

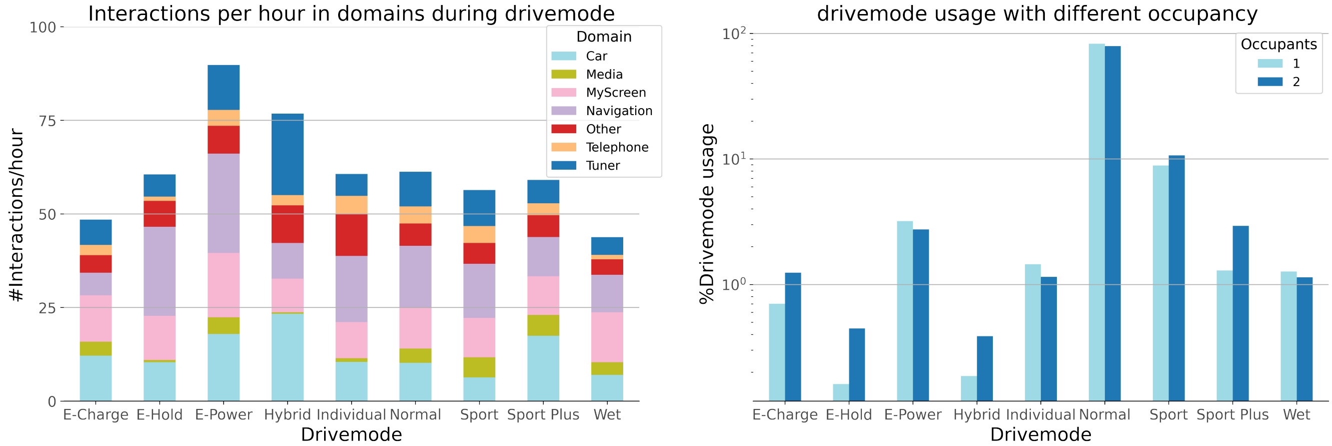
Drive mode	**1 Occupant**	**2 Occupants**
E-Charge	32	34
E-Hold	7	12
E-Power	146	77
Hybrid Auto	9	11

Table 4: Drive mode usage in hours per occupancy

Drive mode	Description	Car
Normal	Initial drive mode when starting the car, normal car settings	911, Panamera
Sport	Dynamic drive mode for sporty driving; sport exhaust, sport chassis	911, Panamera
Sport Plus	Dynamic car settings optimized for track performance	911, Panamera
Wet	Drive mode to drive on more wet and adverse conditions	911
Individual	Customizable drive mode which can combine characteristics of all drive-modes	911, Panamera
E-Charge	Charges battery with the combustion engine	Panamera
E-Hold	Switch to full combustion, to save battery level	Panamera
E-Power	Full electric drive mode, additional power provided by combustion engine when needed	Panamera
Hybrid (auto)	Switch automatically between electric and combustion	Panamera

Table 5: Description of all drivemodes available in Porsche vehicles

Figure 11a shows the distribution of interactions over domains during different drive-modes, while normalised to the interactions per hour in that domain and drive-mode in both user groups with one or two occupants. Figure 11b shows the difference between the drive-mode usage when the car is differently occupied.



(a) Interactions per hour grouped per domain during a specific drive mode

(b) Percentage drive mode usage over time grouped by occupancy

Figure 11: Drive-mode analysis

The relation between drive mode and speed can go both ways, the drive mode may be dependent on the speed the car is driving or the speed may be dependent on the drive mode. Figure 12 illustrates the share of drive mode usage per defined speed block. A clear trend can be observed where the sport mode usage increases from 5% of the time to over 20% of the time (300% growth) when driving faster. This trend can also be observed in sport plus mode, the usage increased with the speed from 1% of the time to approximately 10% of the time (900% growth). The normal mode seems to be partly replaced by the more dynamic modes (sport and sport plus) at higher speeds. The One-way-ANOVA test showed a non-significant statistical difference in interactions made during different drive modes over the speed range: F(98, 90) = -0.35, $p = 0.4$.

Figure 13 shows the main used domains while driving with an active route guidance. An observation can be made that during active route guidance over 50% of the interactions happen in the navigation domain. All other domains are being used more without active route guidance. A paired T-test showed a significant effect between the mean number of interactions during active and non active route guidance: r(9350) = 4.296, $p = 0.004$. In total, there were 7,212 recorded hours of non active route guidance and 177 recorded hours of active route guidance.

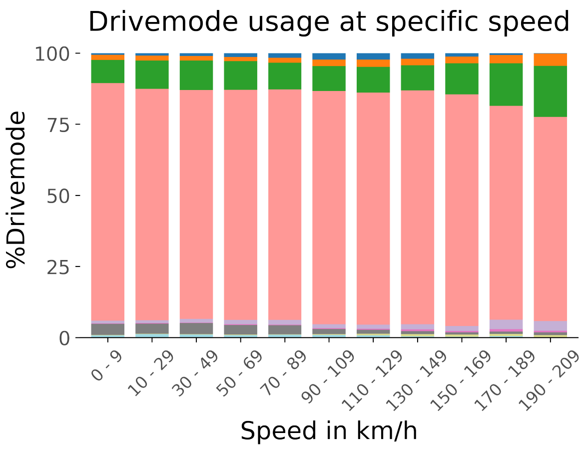


Figure 12: Normalised percentage drive mode usage per speed block

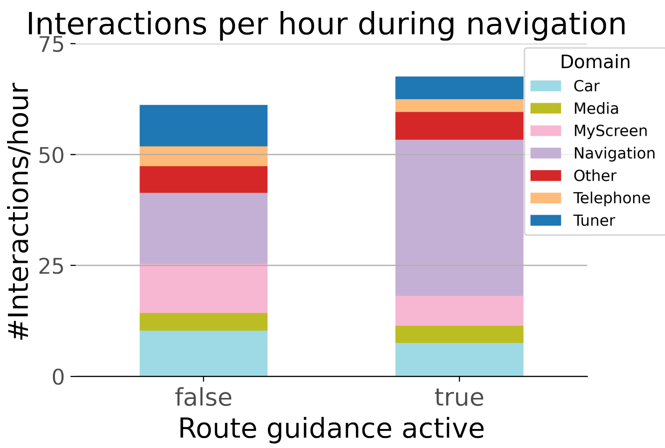


Figure 13: Domain usage in interactions per hour during active and non-active route guidance

Results of the input method analysis are presented in Appendix A. These results do not show any new findings and are omitted in the main part of this report.

5.4 Discussion

The results show some interesting findings to describe the behavior of the drivers. Drives occupied with one occupant show more interactions than drives with two occupants. A possible reason for this finding can be found in Figure 10b. This figure shows that the phone domain is used significantly more with only one occupant than it is when there are multiple occupants in the car. So, communications during drives with one occupant happen through calls via the UI. When driving with multiple occupants, conversations take place with the passengers. A relation can be made between this statement and the introduction, which stated that people are less able to interact with multiple elements in both the car and the UI. So, when people start interacting more with the UI, less interaction is happening with the passengers.

Also, the passengers are taking the initiative in the interaction with the UI when speed becomes higher. Assumed is that high-speed driving uses more cognitive resources from the driver, thus less available for UI interactions. The passenger has almost no extra cognitive load at higher speeds, which is proven by the data.

In the introduction the correlation between interaction behavior and driving behavior is described. Nijboer et al. shows that concurrently performed tasks lead to contention for the resources necessary for the single tasks [31]. This single task includes in this case driving. Based on this finding and the findings stated above, the correlation between driving behavior and interaction behavior is also present in this dataset.

Another aspect of Figure 10b is the navigation domain usage with differently occupied cars. When the car is occupied by two persons, 60% more interaction happen in the navigation domain than with one occupant. This can be explained by the purpose of the drives. The average length of the drive is longer with two occupants than with one occupant. It could indicate that when the car is occupied by two persons, the car is taken for a longer trip, which involves the use of route guidance, contrary to drives with one occupant, which are 30% shorter in duration, that might be used for more commuting purposes. So, if the purpose of the trip is the passenger's purpose, there is automatically a higher likelihood the driver does not know where that is and thus resorts to the navigation system. However, there is no information available about the actual purpose of the trip which could be traced from the route and GPS information.

The statistics from Figure 11a show no correlation between drive modes and interactions in specific domains. Figure 11b shows the usage of these drive modes per occupancy group. Since the values per occupancy group are quite similar, this graph also represents the total drive mode usage. As Table 5 shows, normal drive mode is default which is unchanged for almost 90% of the total time. Consequently, the information about the domain usage in other drive modes is not generalisable and therefore nondescript. For further research, when more non anonymised data becomes available, a better description of the relation between the driving task conditions and the drive mode can be given. In the current analysis, the dataset is composed of Porsche 911 and Porsche Panamera. The Porsche Panamera's in the set are partly hybrid and partly combustion-only. Table 5 shows nine different drive modes, all nine are available in the hybrid vehicle, but only five are in combustion-only vehicles. Since there is no information about the specific car, there is no possibility to verify the exact version. The results of the hybrid drive modes are therefore less reliable due to the lack of available data. This can be solved when the VIN of the cars is registered in the data.

The characteristics of the route guidance, as shown in Figure 13, showed an increased number of interactions in the navigation domain when route guidance is active. Yet, any information about these actions is unknown. It is notable that the data only gives insight if an action has taken place instead of the characteristics of the action. Further research could go deeper into the actual interactions; where, why and what interactions are happening.

The main limitation of the study is the anonymised dataset. With personal information about the driver, the analysis could go to a personal level. Instead of using all the data for the analysis, specific data of each person could be analysed to find the differences or similarities between drivers. These similarities or differences can help understand new drivers based on their personal details and characteristics before any data is collected.

The current analysis is because of the properties of the dataset, applicable to a general group of Porsche drivers. Since not every driver owns a Porsche, the findings of these analysis could also be questions for generalisability. Further research could expand this analysis with similar data from different car manufacturers to compare Porsche drivers with other car brand drivers. This could be the first step towards comparing interaction characteristics among different car brand users.

5.5 Conclusion

The exploratory data analysis is performed to give some insights into the interaction behavior on the UI of the drivers. The analysis showed the correlation between the interaction behavior and specific driving task conditions. The p-values of each variable plotted in the figures above are calculated to verify if there is a significant effect of a certain driving task condition. Table 6 summarises the p-values with the tested hypothesis. The last column of Table 6 shows the important features which could affect the results of the next chapter, to learn the interaction patterns of the drivers with the UI to predict their next actions (Chapter 6).

Hypothesis	P-value	Important feature
Occupancy	<0.001	Occupancy
speed	<0.001	Speed
Domain	<0.001	Current Domain
Route guidance	0.004	Navigation true/false
Drive mode	0.4	None

Table 6: P-value and important features for next steps

The important features which affect the UI interaction behavior significantly, and can therefore add valuable context to the data, are: *Current domain*, *Route guidance*, *Speed* and *Occupancy*.

Based on the presented results of the analysis, the sub-research questions which support the main research questions can be answered.

- The interaction behaviour is influenced by the occupancy. Drivers tend to interact more with the UI when there are no passengers present in the vehicle. When driving faster, passengers tend to take over the UI interactions.
- The interaction behavior is influences by the speed. The total number of interactions decrease as the speed increases.
- The interaction behavior is not influences by the drive mode. Statistical tests have shown that there is no difference in interaction among the different drive modes

6 Next user action prediction

6.1 Introduction

The introduction provided an overview of why it is important to predict the user's next action on the UI. Section 5 shows the exploratory data analysis which describes the correlation between certain driving task conditions and the interaction behavior. The outcome can be used to determine which features should be used in the prediction modeling. To predict the next user action, the dataset should be pre-processed in such a way, that the models can use the data and find the relationship within the history to make a prediction. Suppose the data is split up in n-batches of fixed length T, where the n^{th} batch of sequences can be expressed as equation 2.

$$(\vec{x}^{n}_{1:(t-1)}, y^{n}) = [x^{n}_{1}, x^{n}_{2} ... x^{n}_{t-1}, y^{n}] \tag{2}$$

Where $\mathrm{x}^{\mathrm{n}}_{1:(\mathrm{t}-1)}$ is the input and represents the history of the sequence of batch n where $\vec{x}^{n}_{1:(t-1)} \in \mathbb{R}^{d}$ where d is the dimension of the input space. While y^{n} is the output of the same batch and similar to $\mathrm{x}^{\mathrm{n}}_{t}$. The goal of time series prediction is to predict output y^{n} given the historical sequence $\mathrm{x}^{\mathrm{n}}_{1:(\mathrm{t}-1)}$. The current dataset needs some feature engineering to make it suitable for the machine learning algorithms. Based on the outcome of section 5, the import features necessary for describing context and UI usage behavior are aggregated to form the dataset. So, that the input transfers the most amount of information to the model. Equation 3 shows the features and the actual prediction label.

$$\begin{aligned} \vec{X} = [\text{timestamp}_{1:(t-1)}, \text{action}_{1:(t-1)}, \text{navigation}_{1:(t-1)}, \text{speed}_{1:(t-1)}, \text{media}_{1:(t-1)}], \\ Y = [\text{action}_{t}] \end{aligned} \tag{3}$$

Based on $\vec{X}$ the dataset forms a multi-dimensional one-step-ahead time series prediction problem, where the output, which is only the next consecutive action, is defined based on the (contextual) features at every timestamp. The data is similarly processed as described in section 4, by transforming the on-change data to a continuous dataset describing each feature at every timestamp. Value t describes the amount of history used per batch n, this value t is called the lookback window. Since the sequences can be longer than the lookback window, the lookback window is used as a sliding window, where the window is moved over the sequence, every iteration one timestamp until (y^{n}) becomes the last value of the sequence. When the dataset is transformed into a ready-to-use time series set for prediction purposes, the methods to tackle this problem will be discussed in the following part. The upcoming sections will try to answer the main research question:

Can the next user action be predicted based on the interaction history and specific driving task conditions?

The models described in Section 3 are being used and compared in their performance in predicting the next user action, to answer the research question stated above.

6.2 Method

Since there are a lot of models suitable for the problem described in the related work, we will divide the models in three different categories; statistical, machine learning and deep learning. The following evaluated models were used:

- Statistical models: ARIMA(Vector Autoregressive Moving-Average (VARMA))
- Machine learning models: SVM, RF
- Deep learning: RNN (LSTM)

Firstly, the models are briefly explained below in the order as presented above.

The ARIMA model is introduced in 1970 by Box and Jenkins [22]. The prediction done by the ARIMA model is by expressing it as a linear combination of the current and the past values, as shown in equation 4.

$$Y_t = \sigma_0 + \sigma_1 Y_{t-1} + \sigma_2 Y_{t-2} ... + \sigma_p Y_{t-p} + \mathcal{E}_t - \theta_1 \mathcal{E}_{t-1} - \theta_2 \mathcal{E}_{t-2} - ... - \theta_q \mathcal{E}_{t-q} \tag{4}$$

Where $\mathcal{E}_t$ and Y_{t-p} are, respectively, the random error and the value at that timestep, θ_q and σ_p are the coefficients of the corresponding values and p and q are referred to as the autoregressive and moving average and are integers [24]. The disadvantage with ARIMA is that the input is a single-dimensional vector with the history, so it is not able to take multiple variables at a specific timestamp into account. VARMA, introduced by Wei works in the same way as the ARIMA but can take a multivariate time series history as input [32].

SVM is a regression method which aims to create a linear function from the input $\mathbf{X}_t$ to the output $\mathbf{Y}_t$ as decribed in equation 2. The linear function can be written as:

$$f(\mathbf{X}) = \omega\mathbf{X} + \mathbf{b}) \tag{5}$$

Since the dimension of the input space is in this case 7, linear separation is not possible. Therefore, a kernel function is used. The kernel function maps the input space to a hyperplane to separate the data in the feature space [33]. For multiclass classification several kernel functions can be used for the mapping: Cauchy, Gaussian, Hyperbolic secant, Laplace, Squared Sine, Symmetric, and Triangle [34]. The classification of the mapped space by the kernel is done by using multiple binary classifiers, with the most popular being one-against-all/one-against-other and one-against-one [35]. To separate k-classes, one-against-all trains k classifiers of which each separates 1 class from (k-1) classes. The classifier which generates the most obvious fit determines the label for datapoint $\mathbf{X}$. One-against-one trains k(k-1)/2 binary classifiers. The actual classification is based on aggregating all the outputs of the classifiers.

RF is a machine learning method that combines multiple decision tree algorithms to overcome the problems with single decision tree algorithms: low bias and high variance. These characteristics tend to let the algorithm overfit on noise in the data [36]. This makes RF more robust than decision trees. Because of the multiple decision tree composition of the algorithm, the dataset is split by the number of decision trees the forest is composed of. The decision trees do not see the complete dataset which creates lower variance in the data but increased bias. At every node of the tree, the partition of the data is split again based on the label characteristics of the data. Data is being split repeatedly until there is one outcome. This outcome is matched to the label corresponding to the input data. The quality of the split is being expressed in the Gini impurity index, which is described in equation 6 for node N. The labels with the lowest

Gini index determine the split at each node [37].

$$g(N) = \sum_{i \neq j} P(\omega_i) P(\omega_j) \tag{6}$$

P is the proportion of data with label i/j in relation to the whole dataset. This means that the set is constantly split by the biggest subsets.

As mentioned in section 3 deep learning is becoming more popular due to its performance in several fields. Thus, time series classification is also tackled by deep learning. The basics of deep learning start with neural networks but RNN are especially interesting for this use case. RNN consist usually of three layers: input-, hidden- and the output layer. As the name suggests, the input layer processed the input towards the hidden layer. This hidden layer receives not only the input but also the output of the previous timestep, in this case it is able to take the history into account.

LSTM is a variant of the RNN and is introduced by Hochreiter et al. [38]. Because LSTM solves the exploding gradient problem of the RNN, LSTM is being explained in this part. LSTM can use, carefully selected, time-dependent patterns for the predictions of the next timesteps [39]. LSTMs were hence developed to enable more access to history by improving the gradient flow within the network throughout the timesteps. This is achieved by using the cell state which stores long-term information, modulated through a series of gates as shown in Figure 14

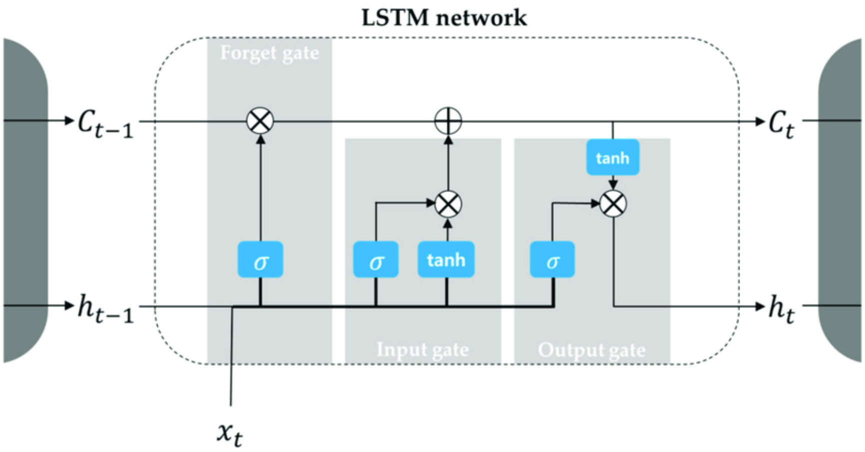


Figure 14: schematic representation of LSTM-network [40]

Figure 14 shows three gates: forget-, input- and output gate. Each gate uses its activation function to use the gate differently. LSTM introduced a function that memorizes what the network predicted in previous timestamps, summarized in the forget gate. The forget gate takes the cell state of step C_{t-1}, output H_{t-1} and the input x_t and decides what to forget and what to pass through to the input gate. Subsequently, the network can remember and forget explicit time steps so it can use a larger amount of history to base the next prediction on. This prediction can then be used for the next time step where it is being processed again: ignored, forgotten or remembered.

6.3 Data preprocessing

To test the methods the dataset needs to be prepared in a way that is appropriate for machine learning algorithms. The data is already converted to a timeseries specific format, where there is

a sequence of data followed by the consecutive action. The set lookback window is determined to be 29. So 29 samples of history (X) and the 30^{th} (Y) is the sample to predict. In total 3.500.000 series were generated by using a sliding window on the data. Some signals, redesign of navigation map in particular, could appear multiple consecutive times (up to 30 consecutive signals). The batches of signals were limited to five similar consecutive signals, since these would not always represent real actions.
When a model is fit or trained to all the samples, the model cannot be tested for generalization. Crossfold-validation is performed to assure this. The data is split in a training, validation and test set (80%/10%/10%). The training set is used for training the algorithm, the validation set is used for monitoring the performance between training and the test set is used for evaluating the performance of the model. Since the test-set is not used for training, the performance on the test set shows the ability to generalize. The complete dataset consist out of 3.500.000 timeseries.

6.4 Evaluation

6.4.1 Evaluation metrics

To compare the aforementioned methods, four different metrics are used to calculate the performance of each method. The following metrics are used: accuracy, weighted precision, weighted recall and weighted F1 score. The metrics are explained in this section. The following abbreviations are used: True Positive (TP), True Negative (TN) for the correct predictions and False Positive (FP) and False Negative (FN) for the incorrected predicted classes.

The accuracy represents the number of correct predictions per total number of predictions. The accuracy is defined by equation 7:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

Precision is the ratio between the number of correct predictions and the number of total predictions off that class. Precision is calculated by equation 8:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{8}$$

Recall is the score which represents the ratio between the correct predictions and the total number of relevant predictions. The equation to calculate the recall is shown in equation 9:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{9}$$

And the F1 score combines the precision and recall by taking the harmonic mean, this measure is shown in equation 10:

$$\text{F1-score} = 2 \cdot \left(\frac{\text{precision} \ \cdot \ \text{recall}}{\text{precision} \ + \ \text{recall}} \right) \tag{10}$$

For balanced datasets, these methods represent perfectly the performance of each model. If the models shows some imbalance, where one class is more present than other, these metrics can be tuned to take the distribution of the classes into account. This is called the weighted-precision, weighted-recall and weighted-F1-score.

6.4.2 Evaluation results

The methods discussed in Section 6.2 are tested and the results are presented below. The models are implemented and fitted to the training set. The (hyper-)parameters and test accuracy are shown below. Every model is optimized using a 10-fold grid-search cross validation method, which tunes the hyper parameters (see Table 7). The results of the 10-fold validation are shown in Table 8.

To show the effectiveness and performance of each model (section 6.2) the prediction results are compared in terms of validation accuracy, precision, recall and F1-score, as defined in section 6.4.1. For effective comparing, the same training and test set is used for the training an evaluation of the models.

Tuning of the models was necessary to use the full potential of the models and improve their performance. This so-called hyper parameter tuning, is done using a grid-search cross validation method. Where every parameter is automatically adjusted to find the best performing setup for every model. The tuned hyper parameters are shown in Table 7.

Model	**Parameter 1**	**Parameter 2**	**Parameter 3**	**Parameter 4**
VARMAX (ARIMA)	Order: 2	Trend: Constant	-	-
Support Vector Machine	Kernel: rbf	Degree: 3	Gamma: 0.012	-
Random Forest	Estimators: 1000	Criterion: Gini	Max depth: ∞	-
LSTM	Layers: 3	Neurons: (400,200,78)	Activation: (Relu, Softmax)	Loss: Cross-entropy

Table 7: Tuned hyper parameters of the prediction models

Table 8 shows the results of every metric per method. The LSTM consistently outperformed every other method in terms of accuracy, precision, recall and F1-score.

Model	**Validation accuracy**	**Precision***	**Recall***	**F1-score***
VARMAX (ARIMA)	59 %	50,1 %	61,3 %	56,4 %
Support Vector Machine	76,9 %	59,4 %	76,9 %	67,1 %
Random Forest	85,3 %	81,5 %	82,7 %	85,3 %
LSTM	87,2 %	84,4 %	87,2 %	85,3 %

Table 8: Performance of prediction models on next action prediction task (*weighted metrics)

6.5 Discussion

The performance of the LSTM has shown that this model is suitable for predicting the next user action. It is able to find deeper connections within the data, where the other models were clearly not able to do that. However, a trade off between efficiency and performance has to be made, since the LSTM has high demands in terms of resources and waiting time. GPU-accelerated computing was necessary which decrease the training time from 20 hours to two hours, while RF and SVM have a training time of 20 minutes (solely on CPU). The feature space used in this analysis (5 features) can be extended for better performance. Additional features can be found by analysing more context related variables in the descriptive analysis.

Since this data is coming from a real life data collection campaign, with a data collector being not always very consistent in terms of performance, the quality of the data could be questioned.

As explained in section 6.3, there has been a pre-processing step where more than 5 consecutive steps were limited to 5 steps. This demonstrates the quality of the data. For future work with a similar data campaign, the signals had to be verified (again) for correctness.

Future research could also increase the number of contextual variables. This work already shows the benefit of adding additional features to the data. When more features become available from the car, this could increase the feature space and gives new opportunities for finding new relations within the data. Other research done at Porsche involved context classification (e.g. roadtype, weather, traffic) based on CAN-signals (e.g. steering input, throttle input etc.). Because this method is based on the same anonymous dataset, with the current privacy regulations, this work could already be implemented with the current models. The more information passed as input for the models, the more information the prediction can be based on.

In the future, when more autonomous data will become available, the autonomous part can be added to the feature space so the model could learn the personal behavior of the driver. The current model, with the current dataset, is able to learn and predict general patterns. Because all sessions are combined into one dataset, individual behavior cannot be accounted for. For an extension of the current model, together with a potential autonomous dataset, a clustering part can be added. This clustering can be applied for creating groups of users based on similar behavior. The data coming from these groups can decrease the variance in the data, to improve the prediction performance. These groups can also be used for setting a better baseline for new car users. Based on their personal data, the new user can be assigned to a specific group to improve the personalised feeling created by the next user action prediction.

6.6 Conclusion

As presented in the evaluation, the LSTM performed the best based on the defined metrics. To answer the research question, stated in the introduction, The LSTM-network can predict the next user action with an accuracy of 87% based on the driving task condition and interaction history. Due to the findings from the descriptive analysis, this performance can be used to develop the next generation adaptive UI. With this adaptive UI we hope to decrease the distractions caused by the in-car infotainment system and thus improve traffic safety for all road users.

7 General conclusion and discussion

7.1 General discussion

This study presents the characteristics of the interaction behavior of Porsche drivers about the context. To be able to present this, several difficulties had to be overcome. Appendix C shows the problem of the access to the data as it is used for this analysis. After granted access, the actual analysis could start and was executed as described in the previous sections. To predict the next user action a descriptive analysis was conducted based on the research questions. The sub-questions were answered by the descriptive analysis in Chapter 5 and the main question has been answered in Chapter 6.

The introduction gave insights in the effect of several driving task conditions such as speed, different occupancy, drive mode and media/navigation usage. These studies had in common that only one condition was checked. This work showed a combination of several conditions combined in one dataset. This approach gives new insights into the interaction behavior, but also helps understand the final predictions coming from the machine learning algorithms.

It is notable that the current dataset only recorded the occurrence of an action instead of the characteristics of that action. Further research could go deeper into the actual interactions; where, why and what interactions are happening. To get a better understanding about the reasons behind a certain interaction. This could help describe the interaction behavior better and therefor create a better foundation for the prediction algorithm.

The work of Wolf et al. showed an interesting approach in using similar user data from the fleet [14]. This work continued their work by including more variables and predicting the next user action for a more practical application. Future research can build on this approach and results. More variables can be analysed and more features can be added to the input space of the prediction algorithms. The prediction model can also be expanded to a more practical-oriented application where it is built into a UI of a driving simulator and adapts the UI based on the predictions. On the subject of practical-oriented applications, the prediction models should be tuned to comply with the physical computational limitations of the vehicle. In this work, no constraints were taken for these limitations, while in practice this might become a problem. In terms of machine learning performance, more work can be put into the prediction algorithm and the tuning to improve the performance of the current models. With the current accuracy of 87%, a new aim could be 95+%.

Nonetheless, a dataset of this scale is still the main contribution of this work. Similar sized datasets have not yet been found in other publications in the automotive industry. Other research mainly collected data in driving simulators. Even though this data is not much different from naturalistic driving data [41], collecting this amount of data in a simulator has also not been done yet.

Collecting this dataset, describing the influencing variables, and predicting the next user action shows not only a new approach to using in-car collected data but also shows a big potential for future research to build on.

7.2 General conclusion

The current study investigated the interaction behavior in naturalistic driving scenarios. The context variables which influence the interactions performed on the UI significantly, were tested. This interaction behavior is modeled by several prediction models to predict the next user action based on this behavior. The findings can be summarised in the following points:

- The occupancy, speed, active route guidance, and current domains influence the interaction behavior significantly.
- A LSTM was able to model the interaction behavior the best, based on an 87% prediction accuracy during cross-validation.

This work can help improve recommendation systems for future adaptive UI's and help decrease distraction from the UI during driving, thus increasing traffic safety.

Besides the stated conclusions, presenting a similar naturalistic dataset from the automotive industry has not been done before. This report shows a small part of the potential use cases for this dataset. Various departments in the automotive industry can adapt these findings for their use cases for a more client-centered process.

7.3 Acknowledgement

Without the help and support of the Porsche employees, this report would not have been possible. Special thanks goes to the EPD department for providing an office, budget and equipment needed for this report. I would like to specifically thank Marco Wiedner, Francesco Branca, Thomas Cros and Enrico Mion (ETH).

Appendices

A Input method analysis

Other plots which were interesting but not showing enough significant findings to be put in the main part of this report include the following: Input method and screennames. Starting with the input method, numerous ways to analyse the input method were discussed followed from the hypothesis that the input method is changing when the speed is increasing. Figure 15 shows the distribution over the input method. The touchpress and touchrelease are dominating here and responsible for 90+% of the total interactions. Therefore, figure 16 shows the same distribution, but without the dominating actions (touchpress/release).

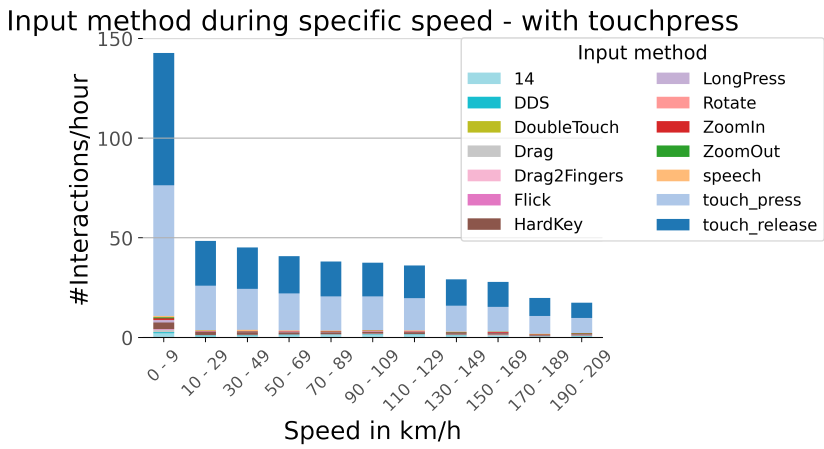


Figure 15: Combined interactions over speed per all input methods

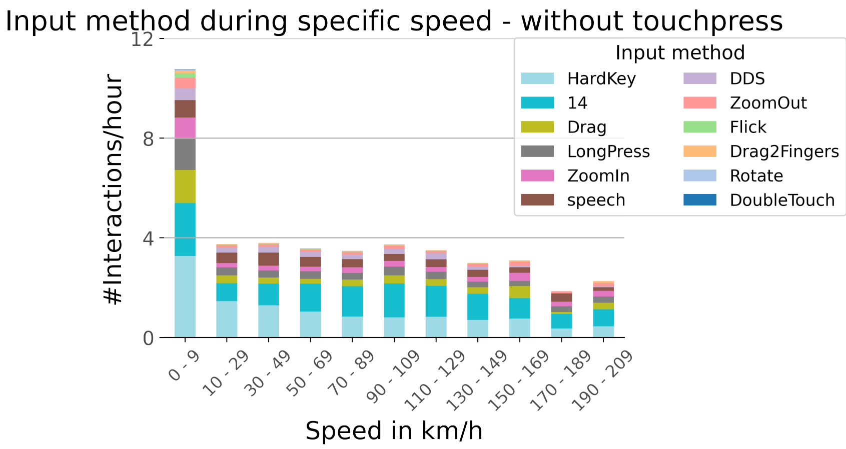


Figure 16: Combined interactions over speed per input method (without touch press/release)

Expected would be a different trend in the inputmethods when the speed increases (e.g. more steeringwheel hardkeys when driving fast). Unfortunately no correlation between speed and input method was found. Another hypothesis tests the statement that a differently occupied car, behaves differently when the speed increases in terms of input method. Expected would be a decrease in touch method for cars with one occupant and an increased usage with a 2+ occupied car in relation to the car with only one occupant. Therefore, figure 17 and figure 18 show the actions done on the infotainment system by respectively a drive with one occupant and with two occupants. Since the share of touch press/release stays over the speed on average 90% for both occupied drives, these actions are ignored.

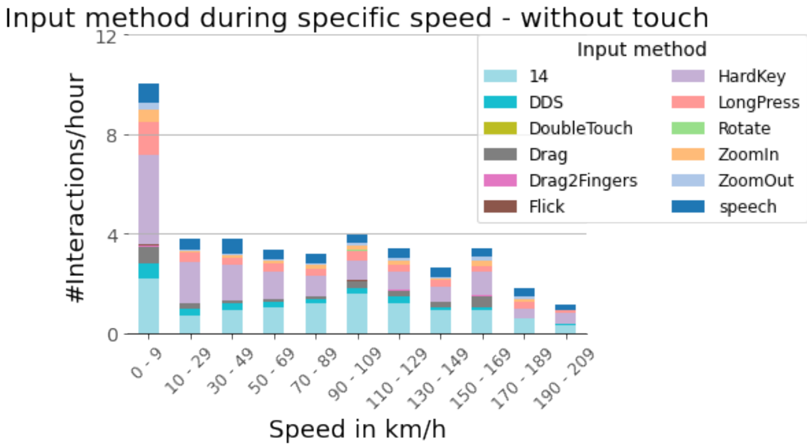


Figure 17: 1 occupant - interactions over speed per input method (without touch press/release)

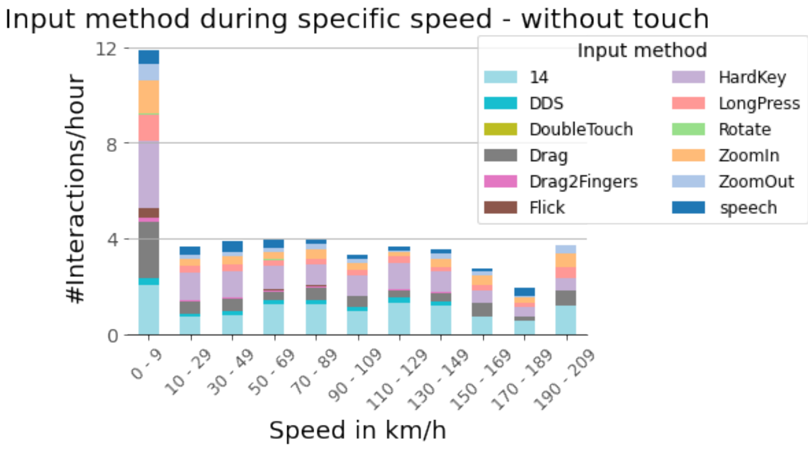


Figure 18: 2 occupants - interactions over speed per input method (without touch press/release)

The actions shown in figure 17 and 18 show a slight difference in navigation map movements e.g. *Drag*, *Drag2Fingers*, *ZoomIn*, *ZoomOut*. These actions can all be related to the domain Navigation, where actions are performed on the map. This is completely in line with the finding presented in figure 10b, where the navigation domain is significantly used more with a two person occupied drive versus one person occupied drive. Since this statement is already been proven, the input method is omitted in the report.

B Screen analysis

In section 5 the interactions are made more specific by relating them to domain the action is performed in. To specify these domains even more, the domains can be split up in the particular screen. Every domain can be divided in a couple of screens, where each screen shows something else related to that specific domain. Table 9, Table 10 and Table 11 show the individual screennames per domain.

Domain	**Navigation**	**MyScreen**
Screenname 1	Navi_Map_REDESIGN'	InfoWidgetMyScreen'
Screenname 2	'Navi_Truffles'	'MyScreen'
Screenname 3	'Navi_DestMain_REDESIGN'	'SET_MAIN'
Screenname 4	'Navi_Truffles_OnlineSearch'	'MyScreenConfigContent'
Screenname 5	'Navi_DestEntryStreet'	'MyScreenInfoWidgetConfigContent'
Screenname 6	'InfoWidgetNoNaviContext_MapRedesign'	
Screenname 7	'Navi_DestEntryCity'	
Screenname 8	'Navi_Dest_LastDest'	
Screenname 9	'_Navi_PoiCategoriesSearchAndResults'	
Screenname 10	'InfoWidgetNaviContextInMain'	
Screenname 11	'Navi_FavList_DestStorage'	
Screenname 12	'Navi_DestEntryHn'	
Screenname 13	Navi_TrafficListOnRoute'	

Table 9: Screennames in Navigation and Myscreen domain

Domain	**Car**	**Tuner**
Screenname 1	DriveControl'	Tuner_Unified_Play'
Screenname 2	'InfoWidgetDriveContex'	'TUNER_BROWSE_FAVS'
Screenname 3	'InfoWidgetAcContex'	'Tuner_Unified_Browse_List'
Screenname 4	'PopRTV'	'Tuner_Unified_Browse_Coverflow'
Screenname 5	'DriveMode'	'Tuner_Dab_InfoWidgetTunerContextRoot'
Screenname 6	'Air992'	'TUNER_GENERAL_SEARCH'
Screenname 7	'Air'	'TUNER_ONLINE_PLAYER'
Screenname 8	'CarSettings'	'TUNER_ONLINE_DATA_END'
Screenname 9	'Set'	'SET_TUNER_UNIFIED_SETTINGS'
Screenname 10	'PopRVC'	'TUNER_ONLINE_GENERAL_STATIONLIST'

Table 10: Screennames in Car and Tuner domain

Domain	Phone
Screenname 1	'Keypad'
Screenname 2	'InfoWidgetPhoneContextRoot'
Screenname 3	'CallListWithDetails'
Screenname 4	'Favourites'
Screenname 5	'CarPlay_Active'
Screenname 6	'NoPhoneConnected'
Screenname 7	'SetPhonePhoneSound'
Screenname 8	'Phone_MIB2p_FAV'
Screenname 9	'SetPhone'

Table 11: Screennames in Phone domain

In all domains certain screens were used significantly more than other screens, just as figure 11 showed with the drivemode. Therefore, the analysis is stopped at this point. The following graphs show the resulting figures by splitting the domains into screens. To simplify the plots, each part only shows one specific domain.

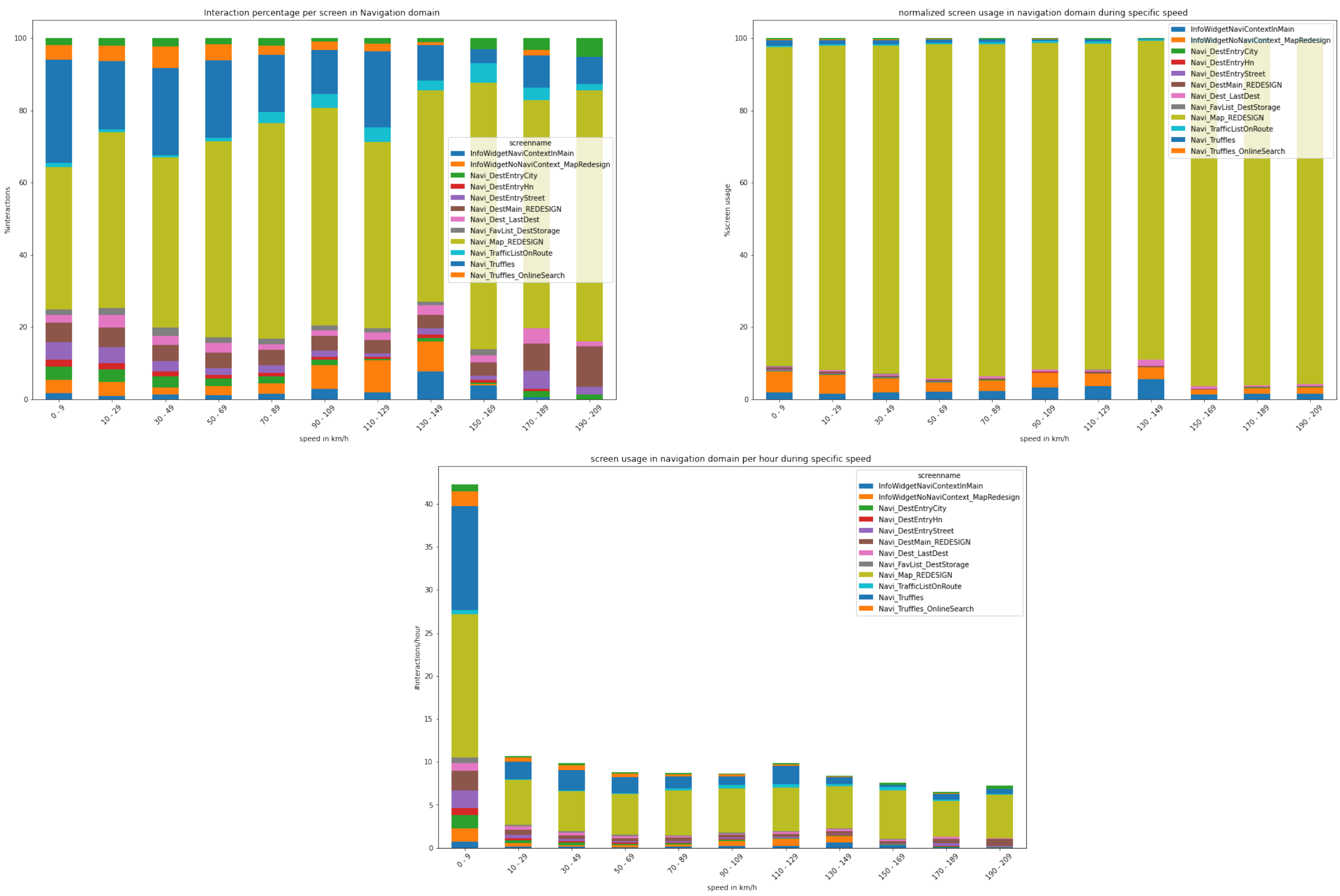


Figure 19: Navigation domain screens per speed group, with normalized interactions, normalized time and interactions per hour

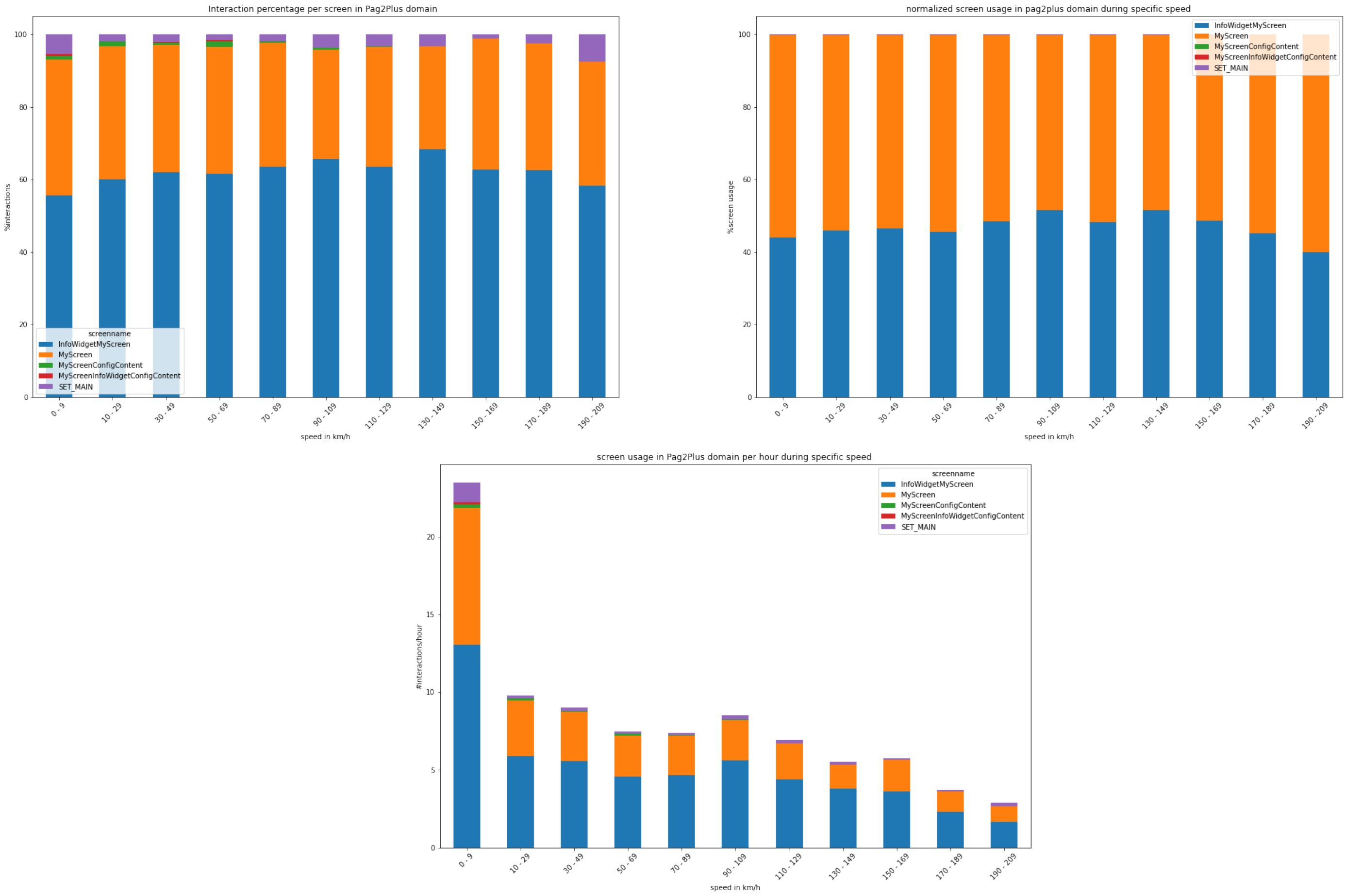


Figure 20: MyScreen domain screens per speed group, with normalized interactions, normalized time and interactions per hour

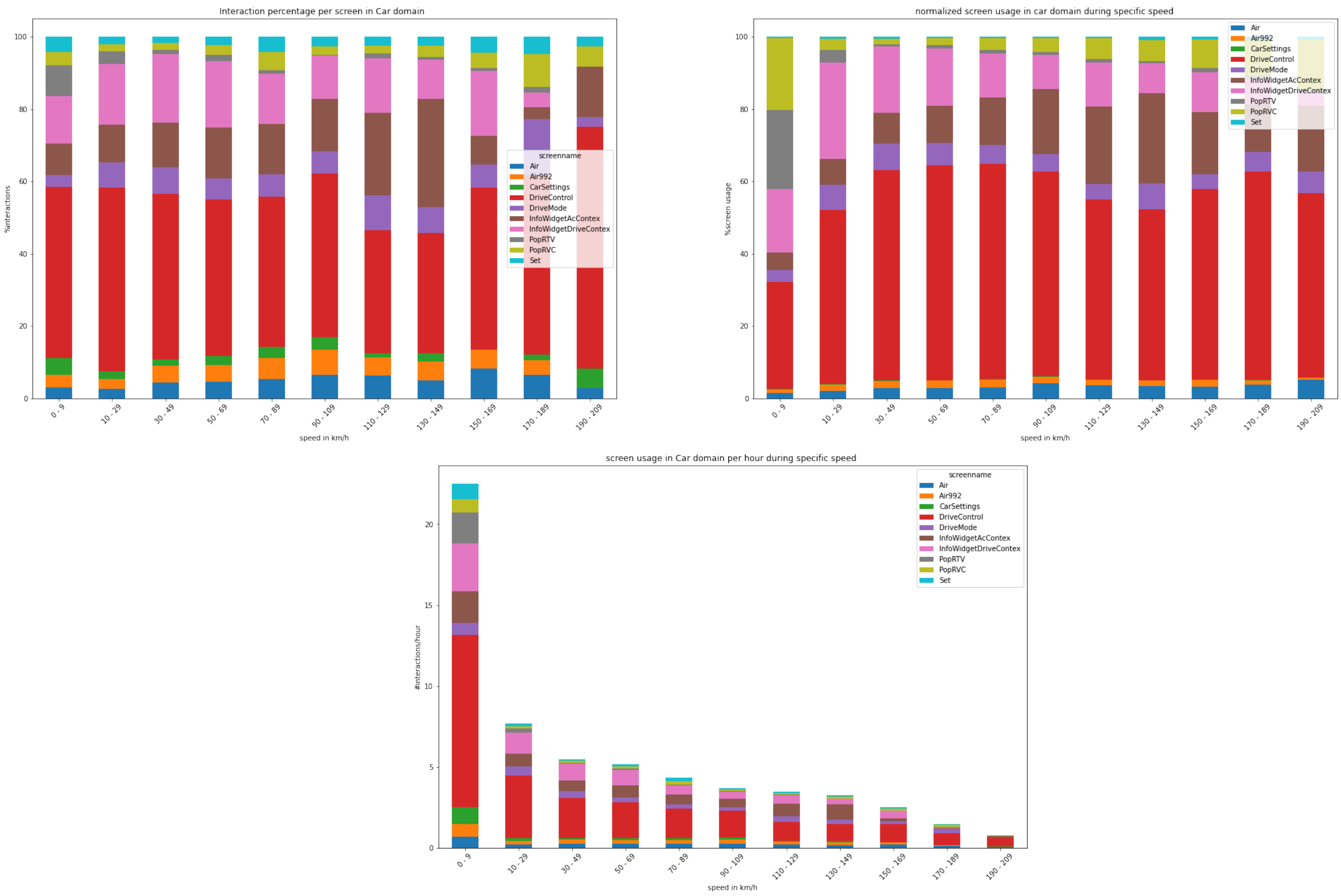


Figure 21: Car domain screens per speed group, with normalized interactions, normalized time and interactions per hour

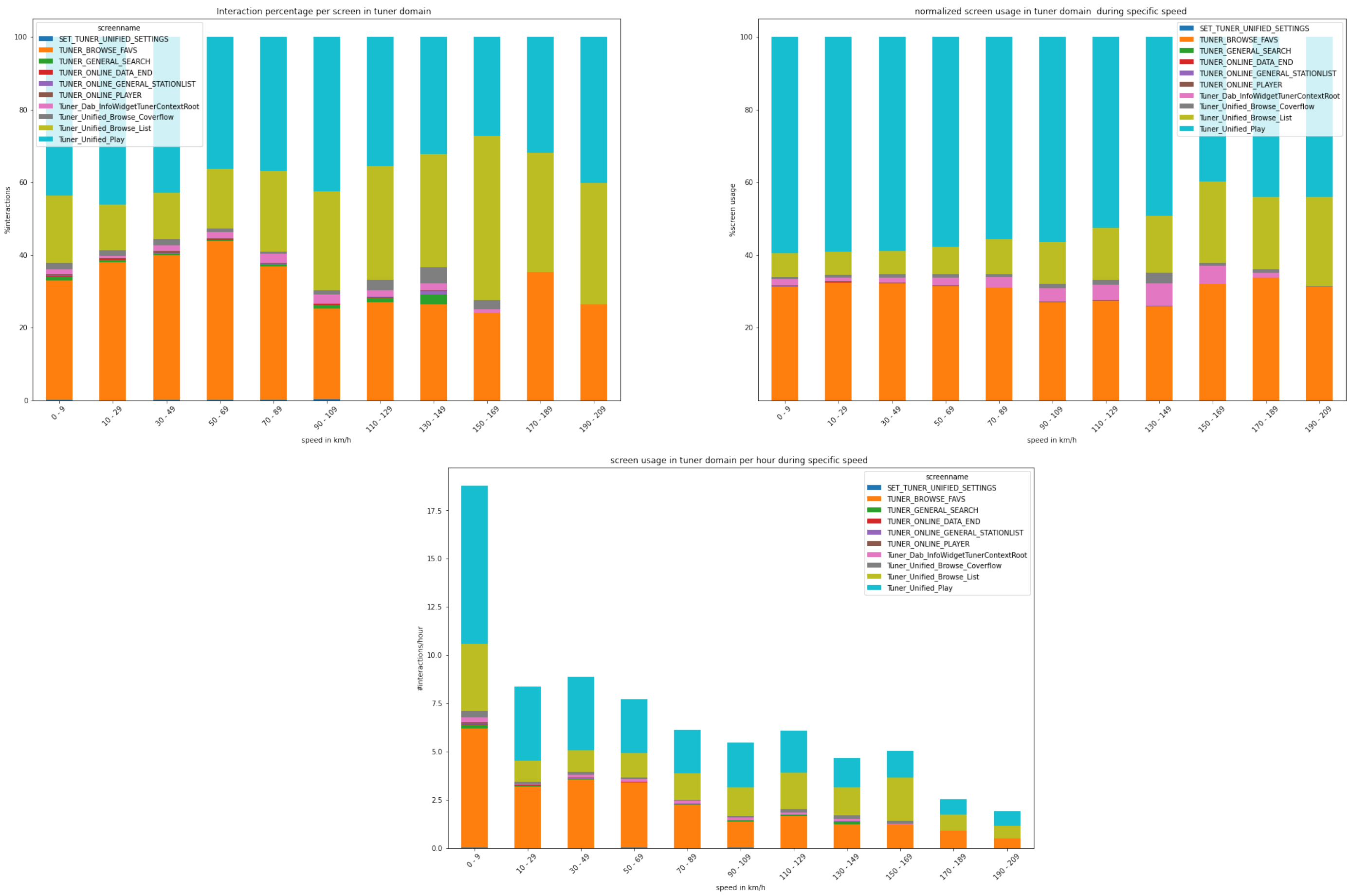


Figure 22: Tuner domain screens per speed group, with normalized interactions, normalized time and interactions per hour

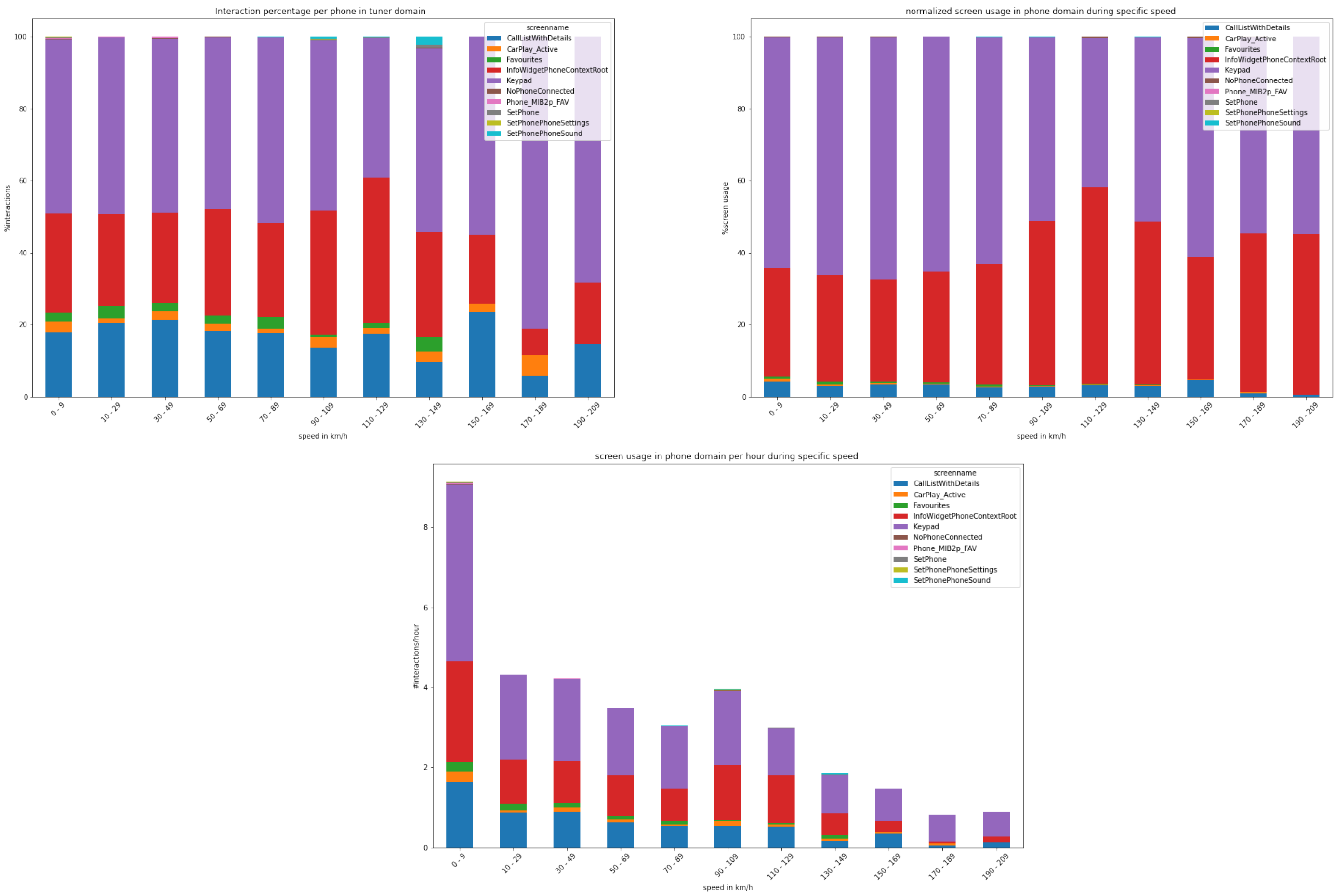


Figure 23: Phone domain screens per speed group, with normalized interactions, normalized time and interactions per hour

C Data collection

As explained in several parts, the data played an essential role in this research. Without the data, it would not have been possible to present the work as it is. However, the actual access to the data took several months. Since this work presents the first usecase for this data within the department, the data had not been accessed before. Since the privacy laws within Porsche limit the access to the data, getting access and processing the data took some effort.

The way to get access to the data turned out to involve only one signature. Because this project was the first of its kind which needed access to the full dataset for research purposes, the responsible departments had difficulties to process this request. To convince the responsible department, several meetings were organized to find similar departments which were using similar data from the same location. It turned out that most departments were using the data for monitoring purposes instead of actually working with the data. After talking with over 5 project teams we got send back to the similar data-owner. The data owner is responsible for the collection of the data, but is not able to give us the access and redirects the request back to the previous teams.

To outbreak this vicious circle a meeting was organized by us with all the departments and persons involved in this problem. During this meeting we presented the exact usecase and purpose. Because all the important people were attending this meeting, a very short discussion was needed to grant the data access for our research.

After the access was processed, the data became visible in the data clusters of Porsche. However, for prediction purposes, GPU-accelerated instances were necessary to process the data. Since these instances on the Porsche cluster did not provide that, the whole cycle started again. Because the access was one thing, but transferring it to another location to do GPU tasks, was another thing.

Luckily, there were some teams involved which establish a connection between the Porsche cluster and AWS. After having several meetings with both teams, it became possible to create a secured connection from the Porsche Cluster to AWS where an instance with GPU and a sufficient amount of RAM could be chosen. From this the prerequisites of the actual analysis were established and the analysis could start.

D Code

In terms of cleanliness all code is bundled and annotated and published on github. The following link shows the repository with all files used for these analyses. a README.md file is shown below to describe the files already in this report.

README

Next user action prediction

To improve traffic safety, focus has been put on developing an adaptive user interface. This adaptive user interface is able to adapt to the needs of the user at a certain point in time given the context. To discover what context is important, real interaction behavior needs to be analysed. Therefore, this repository presents the script necessary to perform a descriptive analysis to analyse what features are important. These features are then used to predict the next user action. \

The following repository contains the files necessary to run:

- Data preprocessing
- Descriptive Analysis
- Data preprocessing prediction
- Deep learning algorithm
- Benchmark models

All files can be run separately. The data frames of all files are included in this repository.

Files

MIB2_preprocess.ipynb uses **combined_csv_48000.csv**

Descriptive_MIB3.ipynb uses **data_mib3.zip**

Descriptive_MIB2.ipynb uses **data_1000_speed.pkl**

DL_LSTM_GPU.ipynb uses **data_1000_speed.pkl**

Baselines_cpu.ipynb uses **data_1000_speed.pkl**

Description

Data preprocessing

MIB2_preprocess.ipynb preprocesses the data to deliver a ready-to-use dataframe for the descriptive analysis. The file cleans the dataset by for example filtering sessions that contain errors and/or sessions unnecessary for the performed analysis.

Descriptive Analysis

Descriptive_MIB2.ipynb uses the dataset from the preprocessing script and forms a run-alone script to perform the descriptive data analysis. The descriptive analysis involves an analysis of contextual variables which influence (or not) the interaction behavior of the UI users.

Descriptive_MIB3.ipynb uses also a preprocessed dataset coming from a different series of cars (equipped with a MIB3+ datacollector). Also, an analysis has been done on this dataset to verify the differences in signals.

Predictive analysis

DL_LSTM_GPU.ipynb performs some processing to the dataset to make it ready for time series prediction usecases. With the processed dataset, an LSTM is trained to predict the next user action.

Baselines_cpu.ipynb is the script that takes care of the baselines to compare the LSTM performance. This script includes the random forest, Support vector machine, and the VARMAX algorithm.

Requirements

And requirements.txt is included in this repository. When using pip: \ `pip install -r requirement.txt`

Python 3.8 is used for all the analyses.

Authors and acknowledgment

All code is written by J.T. Haringsma for finalizing the master in terms of a master thesis. All thanks go to Dr. Ing. hc F. Porsche AG and in particular, EPD where Marco Wiedner supervised this project. From the TU Delft side, Dr.ir. JCF (Joost) de Winter supervised this project, which was highly appreciated.

Disclaimer

References

[1] National Highway Traffic Safety Administration and Us Department of Transportation. *Overview of the 2019 Crash Investigation Sampling System*. Tech. rep. 2019.

[2] National Safety Council. *Understanding driver distraction: How banning use of cell phones and interactive in-vehicle technology while driving can save lives*. Tech. rep. 2020.

[3] Christopher J.D Patten et al. "Using mobile telephones: cognitive workload and attention resource allocation". In: *Accident Analysis & Prevention* 36.3 (May 2004), pp. 341–350. ISSN: 00014575. DOI: 10.1016/S0001-4575(03)00014-9.

[4] Manfred Tscheligi and Wesson Janet and Vogts Dieter. *The Impact of an Adaptive User Interface on Reducing Driver Distraction*. 2011.

[5] Oscar Oviedo-Trespalacios et al. "Can our phones keep us safe? A content analysis of smartphone applications to prevent mobile phone distracted driving". In: *Transportation Research Part F: Traffic Psychology and Behaviour* 60 (Jan. 2019). ISSN: 13698478. DOI: 10.1016/j.trf.2018.11.017.

[6] Gregory M Fitch et al. "The impact of hand-held and hands-free cell phone use on driving performance and safety-critical event risk ". In: (2013).

[7] John H.L. Hansen et al. "Driver Modeling for Detection and Assessment of Driver Distraction: Examples from the UTDrive Test Bed". In: *IEEE Signal Processing Magazine* 34.4 (July 2017), pp. 130–142. ISSN: 1053-5888. DOI: 10.1109/MSP.2017.2699039.

[8] Chen Chai et al. "An Automated Machine Learning (AutoML) Method for Driving Distraction Detection Based on Lane-Keeping Performance". In: (Mar. 2021).

[9] Cheryl Falk. *Exploring the UI Universe: Different Types of UI*. Sept. 2014.

[10] J. W. Senders. " Visual sampling processes." In: (1983).

[11] T. Melman et al. " Do sport modes cause behavioral adaptation?." In: (2021).

[12] John D. Waugh et al. "Cognitive Workload while Driving and Talking on a Cellular Phone or to a Passenger". In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 44.33 (July 2000), pp. 6–276. ISSN: 2169-5067. DOI: 10.1177/154193120004403354.

[13] Kenneth Majlund Ba h et al. "You can touch, but you can't look". In: *Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems - CHI '08*. New York, New York, USA: ACM Press, 2008, p. 1139. ISBN: 9781605580111. DOI: 10.1145/1357054.1357233.

[14] Jannik Wolf et al. "HMInference: Inferring Multimodal HMI Interactions in Automotive Screens". In: *13th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. New York, NY, USA: ACM, Sept. 2021. ISBN: 9781450380638. DOI: 10.1145/3409118.3475145.

[15] Jiqiang Song, Eugene Y. Tang, and Leibo Liu. "User Behavior Pattern Analysis and Prediction Based on Mobile Phone Sensors". In: 2010, pp. 177–189. DOI: 10.1007/978-3-642-15672-4{_}16.

[16] Iqbal H. Sarker. "A machine learning based robust prediction model for real-life mobile phone data". In: *Internet of Things* 5 (Mar. 2019), pp. 180–193. ISSN: 25426605. DOI: 10.1016/j.iot.2019.01.007.

[17] Hosub Lee, Young Sang Choi, and Yeo-Jin Kim. "An Adaptive User Interface Based onSpatiotemporal Structure Learning". In: (2011), pp. 118–124.

[18] Seckin Karasu and Aytac Altan. "Recognition Model for Solar Radiation Time Series based on Random Forest with Feature Selection Approach". In: *2019 11th International Conference on Electrical and Electronics Engineering (ELECO)*. IEEE, Nov. 2019. ISBN: 978-605-01-1275-7. DOI: 10.23919/ELECO47770.2019.8990664.

[19] P. J. Moore, T. J. Lyons, and J. Gallacher. "Random forest prediction of Alzheimer's disease using pairwise selection from time series data". In: *PLOS ONE* 14.2 (Feb. 2019). ISSN: 1932-6203. DOI: 10.1371/journal.pone.0211558.

[20] Tomislav Hengl et al. "Random forest as a generic framework for predictive modeling of spatial and spatio-temporal variables". In: *PeerJ* 6 (Aug. 2018). ISSN: 2167-8359. DOI: 10.7717/peerj.5518.

[21] Nicholas Sapankevych and Ravi Sankar. "Time Series Prediction Using Support Vector Machines: A Survey". In: *IEEE Computational Intelligence Magazine* 4.2 (May 2009). ISSN: 1556-603X. DOI: 10.1109/MCI.2009.932254.

[22] George Box and GM Jenkins. "GM Jenkins Time Series Analysis: Forecasting and Control". In: (1970).

[23] Arul Earnest et al. "Using autoregressive integrated moving average (ARIMA) models to predict and monitor the number of beds occupied during a SARS outbreak in a tertiary hospital in Singapore". In: *BMC Health Services Research* 5.1 (Dec. 2005). ISSN: 1472-6963. DOI: 10.1186/1472-6963-5-36.

[24] Adebiyi A. Ariyo, Adewumi O. Adewumi, and Charles K. Ayo. "Stock Price Prediction Using the ARIMA Model". In: *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*. IEEE, Mar. 2014. ISBN: 978-1-4799-4922-9. DOI: 10.1109/UKSim.2014.67.

[25] Domenico Benvenuto et al. "Application of the ARIMA model on the COVID-2019 epidemic dataset". In: *Data in Brief* 29 (Apr. 2020), p. 105340. ISSN: 23523409. DOI: 10.1016/j.dib.2020.105340.

[26] Geoffrey Hinton et al. "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups". In: *IEEE Signal Processing Magazine* 29.6 (Nov. 2012). ISSN: 1053-5888. DOI: 10.1109/MSP.2012.2205597.

[27] Olga Russakovsky et al. "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision* 115.3 (Dec. 2015). ISSN: 0920-5691. DOI: 10.1007/s11263-015-0816-y.

[28] Sreelekshmy Selvin et al. "Stock price prediction using LSTM, RNN and CNN-sliding window model". In: *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, Sept. 2017. ISBN: 978-1-5090-6367-3. DOI: 10.1109/ICACCI.2017.8126078.

[29] Sima Siami-Namini, Neda Tavakoli, and Akbar Siami Namin. "A Comparison of ARIMA and LSTM in Forecasting Time Series". In: *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, Dec. 2018. ISBN: 978-1-5386-6805-4. DOI: 10.1109/ICMLA.2018.00227.

[30] Youru Li et al. "EA-LSTM: Evolutionary attention-based LSTM for time series prediction". In: *Knowledge-Based Systems* 181 (Oct. 2019). ISSN: 09507051. DOI: 10.1016/j.knosys.2019.05.028.

[31] Menno Nijboer et al. "Driving and Multitasking: The Good, the Bad, and the Dangerous". In: *Frontiers in Psychology* 7 (Nov. 2016). ISSN: 1664-1078. DOI: 10.3389/fpsyg.2016.01718.

[32] William W.S. Wei. *Time Series Analysis*. Oxford University Press, Mar. 2013. DOI: 10.1093/oxfordhb/9780199934898.013.0022.

[33] Min-Wei Huang et al. "SVM and SVM Ensembles in Breast Cancer Prediction". In: *PLOS ONE* 12.1 (Jan. 2017). ISSN: 1932-6203. DOI: 10.1371/journal.pone.0161501.

[34] Arindam Chaudhuri, Kajal De, and Dipak Chatterjee. "A Comparative Study of Kernels for the Multi-class Support Vector Machine". In: *2008 Fourth International Conference*

on Natural Computation. IEEE, 2008. ISBN: 978-0-7695-3304-9. DOI: 10.1109/ICNC.2008.803.

[35] S. Knerr, L. Personnaz, and G. Dreyfus. "Single-layer learning revisited: a stepwise procedure for building and training a neural network". In: *Neurocomputing*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1990. DOI: 10.1007/978-3-642-76153-9{_}5.

[36] Luckyson Khaidem, Snehanshu Saha, and Sudeepa Roy Dey. "Predicting the direction of stock market prices using random forest". In: (Apr. 2016).

[37] Leo Breiman. "Random Forests". In: *Machine Learning* 45.1 (2001). ISSN: 08856125. DOI: 10.1023/A:1010933404324.

[38] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (Nov. 1997). ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735.

[39] Aditi Mittal. *Understanding RNN and LSTM*. Oct. 2019.

[40] Jeonghun Choi and Seung Jun Lee. "Consistency Index-Based Sensor Fault Detection System for Nuclear Power Plant Emergency Situations Using an LSTM Network". In: *Sensors* 20.6 (Mar. 2020). ISSN: 1424-8220. DOI: 10.3390/s20061651.

[41] Jamil Hussain et al. "Model-based adaptive user interface based on context and user experience evaluation". In: *Journal on Multimodal User Interfaces* 12.1 (Mar. 2018). ISSN: 17838738. DOI: 10.1007/s12193-018-0258-2.