Proposal of an adaptive infotainment system depending on driving scenario complexity.

Miguel Angel Galarza Osio

A thesis presented for the degree of
Doctor of Philosophy

Project founded by SEAT, UPC and Catalonia government.

September 19, 2018

A mis padres, por enseñarme valores fundamentales cómo respeto, tolerancia, responsabilidad y constancia. A mi pareja, por todo su apoyo brindado durante las etapas mas dificiles, así como por ser una excelente oyente y asesora. A mis amigos, por mostrame las diferentes perspectivas desde la cuales se pueden enfocar los problemas para encontrar soluciones.

Acknowledgements

This study was undertaken as part of an industrial doctorate project founded by three organizations; an academic institution ”Universitat Politècnica de Calalunya”, an automotive company ”SEAT” and a government administration ”Generalitat de Catalunya”. As so, I would like to firstly give thanks to these parts that gave rise to the development of this research.

During the development of the project, many were the people who contributed and gave me support. I would like to express my sincere gratitude to my advisor Dr. Josep Paradells for the continuous support of my Ph.D study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis.

Most of the development was done in SEAT facilities, where a wonderful person, Teresa Bayona, was the one who offered an large support and guide during the undertaking of the project. I would like to thank her for all the support and motivation. I would also like to acknowledge Victor Baños who was the initial responsible of the research and provided me with important orientation to give shape to the project.

I would also like to thank the many co-workers who helped me in different ways. Without their participation and input, the research could not have been successfully conducted.

Abstract

Objective: Implementation and evaluation of an adaptive infotainment system for in-vehicle use which adjusts the user interfaces depending on the current driving complexity. **Motivation:** Integrate new emerging services and functionalities in the vehicle regarding road safety and driving experience. **Methodology:** A predictive model for estimating the driving complexity in real time was created. The prediction is based on data mining techniques using real driving data; the estimation value was then used to adapt the user interfaces based on stated design principles. Finally, it was executed a performance and acceptance test to evaluate the system benefits and integration feasibility. **Conclusions:** It has been demonstrated the feasibility of implementing an adaptive infotainment system. As well it has been received a positive feedback regarding driver's acceptance to this kind of systems.

Keywords

Driving Complexity, Adaptive HMI, Data Mining, Machine Learning, Design Principles, Infotainment.

Contents

List of Figures

List of Tables

List of Acronyms

ANN	Artificial Neural Networks
ACC	Adaptive Cruise Control
A-HMI	Adaptive Human Machine Interface
API	Application Programming Interfaces
BAP	Bedienund AnzeigeProtokoll
CAN	Controlled Area Network
CoAP	Constrained Application Protocol
DDA	Dynamic Decay Adjusment
DT	Decision Tree
DVE	Driver Vehicle Environment
ECU	Electronic Control Units
EEG	Electroencephalography
ESoP	European Statement of Principles on human machine interface
EXLAP	Extensible Lightweight Asynchronous Protocol
FCC	Forward Collision Control
FFM	Five Factor Model
GUI	Graphic User Interface
HGV	Heavy Good Vehicles
HMD	Head Mounted Display
HMI	Human Machine Interface
HRV	Heart Rate Variability
HTAoD	Hierarchical Task Analysis of Driving
HUD	Head Up Display
HyperNEAT	Hyper NeuroEvolution of Augmenting Topologies
IVIS	In-Vehicle-Information-Systems
KDD	Knowledge Discovery in Database
MAE	Mean Absolute Error
MLP	Multi-Layer Perceptron
NHTSA	National Highway Traffic Safety Administration

NN	Neural Network
OSGI	Open Services Gateway initiative
OSM	Open Street Map
PDT	Peripheral Detection Task
PGR	Psycho-Galvanic Reflex
PNN	Probabilistic Neural Networks
POI	Point of Interest
RBF	Radial Based Function
REST	Representational State Transfer
RF	Random Forest
RSE	Root Square Error
RSME	Rating Scale Mental Effort
SA	Situation Awareness
SSR	Subjective Self-Reports
SVM	Support Vector Machine
SWAT	Subjective Workload Assessment Technique

Chapter 1

Introduction

The present study makes a contribution to the field of Human Machine Interfaces (HMI) for In-Vehicle Information Systems (IVIS). When it comes to the design of user interfaces oriented to be used in driving scenarios, two factors of high relevance stand up: driver experience and road safety. In this regards, the main purpose of the research is to propose a set of design principles that balance these two counterparts. As starting point, the existing problematic that triggered the study is presented and the motivation of this dissertation is exposed. As a final point, the conclusions about the proposed methodology for balancing these two counterparts are analysed and the lessons learned are exposed.

Over the last years, a great amount of services easily accessible to users have been developed. In consequence, the world we live in has substantially evolved to a highly connected environment in terms of information. Thanks to the advances in electronics and telecommunications it is possible to get real time information about the surrounding environment everywhere and at any time. This is conceivable thanks to a large deployment of access points, servers, sensors and front-ends distributed everywhere.

From a user perspective, the access to these services is possible through an abundant amount of electronic devices: smartphones, PCs, tablets, wearables, TVs, and vehicle systems just to cite some examples. The introduced network of *Internet of Things* is expanding at a fast rate and as result; it is expected that more and more on-line services could be delivered every day for enriching people's quality of life.

With all this information available, people are getting used to interact with services anywhere and very often. Individuals have been acquiring certain dependency on technology and a need to be always "on-line". Beyond the different positions about the benefits, drawbacks or origin of this phenomenon, it is clear that the business models have to change and adapt their strategies to cope these demands.

Electronic companies, maybe responsible of this phenomenon, are constantly creating innovative services that ease the access to this global network. The success of services translated in form of applications is given by a set of several features as interface design, usability, fashion response time, social impact and real contribution over available services.

Vehicle manufacturers are also aware of this new market and have begun to provide better connectivity capabilities and more powerful hardware that allows users to maintain connectivity while driving. In a near future, electronic systems installed in vehicles will provide every kind of services related to entertainment; vehicle maintenance, driving assistance, environmental alerts, social networks and some other large set of features oriented to improve the driver experience. This big collection of services and functionalities is called infotainment system (portmanteau word constructed by information and entertainment).

Automotive manufacturers have been discerning which business model is the best suited to follow the rapid development pace of services and connectivity mechanisms. So far, the approach followed has been to integrate the infotainment systems in an embedded hardware and install the complete software directly in this hardware. In this model all the functionalities and services are provided by this infotainment system.

Portable devices, on the other hand, are very dynamic and its applications and services are developed and updated very frequently. New vehicle developments requires a larger time than mobile devices and applications given the many requirements that must be fulfilled to ensure a correct integration among the different electronic parts in the vehicle. Moreover, people do not usually buy a new vehicle as often as it does with portable devices. As a result, hardware vehicle infotainment becomes obsolete quite rapidly. This expertise has shown to vehicle manufacturers the potentiality of portable devices and the need of mechanisms to integrate these into the vehicle. As a second solution, vehicle auto-makers are starting to incorporate infotainment systems that can be easily updated trough Over The Air mechanisms.

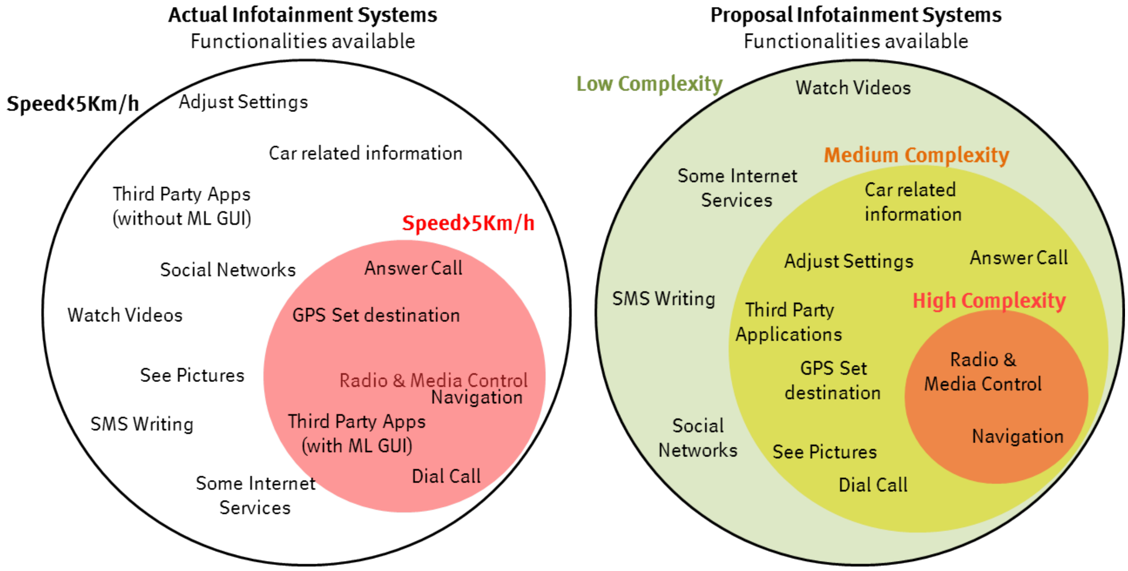
In order to merge the two big worlds of portable devices and vehicle systems, a well-defined architecture where is obtained the best of each one must be established. Portable devices can be used as additional interfaces, sensors, connectivity interface and extensions of processing power, while vehicle systems can provide travel information, driving variables, car status and better distributed interfaces regarding the driving activity. Some connectivity mechanisms developed to mix these two worlds were defined by car-manufacturers, standardized consortium and mobile companies. Some examples are Mirror Link created by the Car Connectivity Consortium (CCC), Android Auto developed by Google and Car Play created by Apple.

Nevertheless, the integration is not as straightforward at it seems when services are expected to be used in a driving context. The incorporation of functionalities in a vehicle must be careful planned since the design must not only guarantee a good driving experience, but also and more important, avoid distractions that may lead to possible accidents. The design of user interfaces for vehicles is based on principles derived from empirical studies and experience. The most recognized works regarding safe designs were advised by countries' national institutions, and consist of rules called *Guideline Distraction Rules*. These standards expose design patterns and general recommendations to ensure proper style to display information and access to interfaces.

Typically, new applications are initially developed to be used in smart-phones or computers. Therefore, when integrated into vehicles a high adaptation of the user interface is made and the applications sometimes lose features, not only related to the design appearance but also to the limitation or restriction of some functionalities. Current vehicle infotainment systems take safety measures consisting of restricting complex services from the point of view of user interaction when the vehicle is travelling at a higher speed of 5 km/h. These limitations affect the user experience and degrade the real purpose of the application.

Aside from the complexity and volume of services provided, a modern vehicle can also have a complex physical architecture constituted by several user interfaces. Old vehicles had only two simple panels to display information: a dashboard that showed the speed and alerts, and a system to control some functions of the radio. Current vehicles incorporate more dynamic and complex interfaces as the instrument cluster (Kombi), Head Unit (HU) and Head Up Displays (HUD). Moreover, some vehicles have already implemented mechanisms to integrate wearable and smartphones as additional interfaces.

Figure 1.1: Restriction of functionalities while driving in actual systems and proposed system.



The interaction with applications varies in complexity depending on the mode used to transmit commands to the system: voice, touch buttons, touch interfaces, gestures; and sensory channels through which the user receives information: visual, haptic or audible. Supposing a driving scenario in which multiple interfaces and functionalities are available for the driver, each one presenting different information, it is clear that the driver can be overloaded, especially if the driving scenario is complex. This leads us to the question of how much is able to process the driver before it increases the accident risk or degrades the driver experience.

In order to provide the increasing amount of services to drivers while ensuring a high level of road safety, a system capable of managing the interaction with functionalities and interfaces must be designed. This system has to take into account not only the current driving complexity but also the user profile and user preferences in order to maintain a good driving experience any time. As previously exposed, actual infotainment systems adjust their behaviour depending only on the speed, e.g, when the vehicle overpasses 5km/h some functionalities are restricted, which represents an excessive restriction for some driving scenarios of low complexity. A proposal to this issue could be the adjustment of functionalities depending on more than two complexity categories. Figure 1.1 shows a graphical representation of the concept.

The main purpose of this dissertation is to evaluate the impact, benefit and suitability of implementing an adaptive infotainment system which provides a dynamic access to functionalities. The proposed system would be responsible of distributing the information among the available interfaces as well as adapting its graphical content and interaction mode depending on the actual driving complexity. This thesis undertakes the study of the technical requirements needed for the implementation and presents a drivers' evaluation regarding their perceptions of safety and user experience.

The hypothesis of the study underlines as follows:

"If we are able to monitor different parameter that allows computing an estimation of the driving complexity, and use this indicator for adapting what, when and how to show and

interact with the information in multiple interfaces, we may reduce driver workload and increase driving experience."

In order to validate the hypothesis, the study was divided in two main phases: (1) the definition and validation of a predictive model for estimating the current driving complexity, and (2) the definition and integration of an adaptive system based on proposed design principles. Both studies were conducted and evaluated in real driving scenarios using data from real vehicles.

The predictor model used for estimating the driving complexity was constructed using machine learning methods and data mining techniques. This approach guarantees that the system can be easily updated and improved. The model was trained taking real driving variables as inputs and drivers' opinions about the complexity of scenario and response times to random events as feedbacks. Different machine learning methods were compared in order to select the most suitable for this purpose.

The adaptive HMI was shaped taking as baseline previously proposed guidelines and recommendations for designing user interfaces. Studies in regards of dynamic and adaptive interfaces are scarce and therefore some design principles were proposed using the background available. The proposed system adapts the user interface based on the estimation of driving complexity, consisting of an indicator of four values.

Once the system was developed, it was integrated in a vehicle and several evaluation tests were carried out in real driving scenarios. Some of the tests were devoted to evaluate the system performance, but most were oriented in evaluating drivers' opinions and perceptions on the system, the implications on road safety and overall acceptance.

The document is structured as follows: Some background is first presented with regards to the evolution and trend of infotainment systems. Similarly, some mechanism previously employed for integrating new services in vehicles and guarantee road safety are exposed. An overview of driver capabilities and limitations is exposed, as well as some statistical information that contributes with empirical data about crash risks is presented. A subsection that summarizes some of the most remarkable machine learning methods is also added, these mechanisms are used during the creation of the predictive model of driving complexity that is integrated in the final system.

Next, the procedure to be followed for finding the most relevant variables associated with driving complexity is presented, and different machine learning methods (models) for predicting unknown scenarios are evaluated. Using this reference, a proposal of adaptive infotainment system is given and usability and acceptance test are carried out to determine the real benefits of the system. The last section presents a set of design principles that can be used as recommendations for integrating new interfaces in a vehicle. Also, the integration of a Head Mounted Display is evaluated in terms of acceptance and visual workload.

Each section includes a short summary containing the main ideas of that part in order to guide the reader trough the concepts exposed in each chapter.

Chapter 2

State of the Art

This section is devoted to introduce some background about the implications of interacting with infotainment systems in driving scenarios. For this purpose, it becomes necessary to understand how drivers perceive their surrounding environment, what they expect from an infotainment system, what methodologies are currently applied in the design of infotainment systems, and how this expertise can be used for improving actual infotainment systems in regards of offering a proper balance between road safety and user experience.

The chapter is divided into four sections which can be seen as an initial step to answer these questions. First, infotainment systems are introduced showing its history, purpose, impact on drivers and new trends. This section additionally provides some technical background regarding vehicles connectivity mechanisms, nomadic device integration and procedures followed in the design of infotainment systems.

Secondly, the driver is defined in terms of their expectations, physiological responses under different driving scenarios and psychological behaviour when driving. This section addresses the estimation of driver workload through different methodologies. The understanding of drivers capabilities and expectations is one of the more important aspects when designing HMI systems. Depending on their workload and preferences, the system should manage differently the interaction between user and interfaces.

The third section summarizes the most notable works formulated in the area of infotainment technologies and HMIs regarding adaptive interfaces. This section holds the baseline required for the statement of new design principles based on dynamic interfaces. Moreover, the evaluation of previous works is essential for understanding the current problematic and what improvements can be made.

Finally, some theoretical background regarding data mining techniques and machine learning method is given. As will be explained later; predictive techniques were broadly used during the study for estimating driving complexities. In this regards, the techniques employed are introduced highlighting the advantages and disadvantages of each one at the moment of estimating outputs.

2.1 Infotainment Systems

2.1.1 Evolution and trends

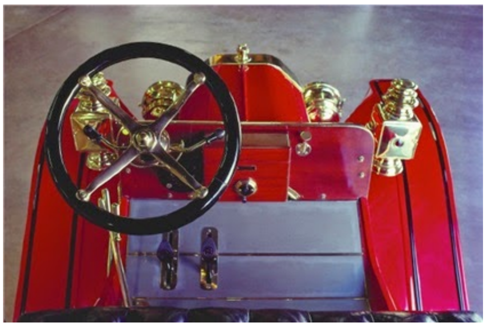
The first vehicle considered of commercial production was invented by Karl Benz in 1886, in those days the vehicle was an electric-mechanical machine without any electronic components. The only role of the vehicle was that for which it was created, transporting passengers from one place to another. The commercialization of vehicles led to a significant change in society that would transform the dynamics of people's life.

Vehicles at that time did not have entertainment systems; nevertheless, information systems that reported data about the trip and vehicle variables were available almost from the beginning. The speedometer and odometer were introduced in some vehicles around 1900s and the fuel gauge in 1914 (Laukkonen, 2013). The information presented to the driver was simple, but it brought some initial changes in the activity of driving.

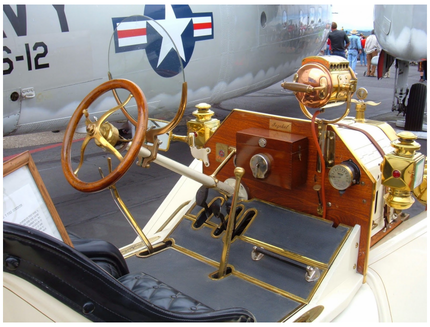
With the start of radio transmissions around 1920, the concept of the vehicle simply as a mean of transportation began to change. Since the introduction of radio broadcasts, vehicle manufacturers began to figure out the manner of integrating radio receptors in their vehicles. The installation was complicated due the high requirements of electrical power and large physical dimensions of first equipment. Nevertheless, by the 1930's built-in Motorola radios were being established as standard features in all its vehicles. The vehicle had now evolved to provide entertainment features during travels. Information and entertainment capabilities had defined a new driving environment giving place to basic infotainment systems.

The science of electronics as we know it started more than 20 years later, with the integration of transistors in practical applications around 1947. Until then, car infotainment consisted of simple radio features and basic vehicle indicators; the introduction of the transistor pulled down size and prices of devices making affordable new technologies. Radio FM was introduced in 1952, record players in 1955, eight track tapes in 1964 and car stereo in middles 1960s (Laukkonen, 2015). Figure 2.1 shows the evolution of vehicles radios in almost 50 years, since 1909 to 1956.

Figure 2.1: Vehicle infotainment evolution from 1909 to 1956.



(a) Ford Model 'T' Touring Car (1909)



(b) Ford Model 'T' Speedster (1913)

(c) Ford Model A (1920)

(d) Mercedes Benz 300SL (1956)

In 1970 eight-track tapes were replaced by the compact cassette, this format had a great success and was kept simultaneously with the new introduced format Compact Disk (CD) in the 1980s. The last vehicle to come with a tape player was sold in 2010 (Gitlin, 2014). In the XXI century new technologies and wireless connectivity protocols began to emerge and to be incorporated into vehicles as Bluetooth and Wi-Fi, which provided capabilities for connecting portables devices directly to the vehicle.

Infotainment systems available in vehicles experienced a great change in the last century thanks to the incorporation of new services as Global Position System (GPS) Navigation, vehicle status monitoring, reproduction of different audio formats and Internet connectivity. The introduction of GPS Navigation systems, with high precision, in the 2000s brought with it the need of larger screens for representing the map information. These screens were later used for showing more complex services and new functionalities. Further, displays dropped in price and became to be incorporated in dashboards for presenting additional information to drivers.

In the 2010s online services as cloud-based music players and web applications were introduced. Automotive and mobile-phone companies, aware of the high expansion of mobile applications, began to proposed mobile-car connectivity mechanisms for mixing these two words and get the best of each one. Initially, most companies implemented proprietary solutions, the communication protocols commonly used to exchange information between the smartphone and the vehicle were based in USB, Bluetooth or Wi-Fi interfaces. Around 2014, several standard connectivity solutions were proposed; the most noteworthy in this regards were MirrorLink, Android Auto and CarPlay.

MirrorLink is a technology proposed by the Car Connectivity Consortium (CCC) as an updated version of the Terminal Mode developed by Nokia in 2009. This technology offers a connectivity mechanism between smartphones and infotainment systems allowing to replicate the mobile screen on the vehicle Head Unit screen. For doing so, MirrorLink implements a set of standard and non-proprietary protocols as IP, USB, Wi-Fi, Bluetooth, RTP, UPnP and VNC (Car Connectivity Consortium, 2016).

The Car Connectivity Consortium was constituted in 2011 and nowadays it has grown to 100 members representing 70 % of the world's auto-makers and 70% of global smartphone market. Some members belonging to the car-manufacturer area are SEAT, Volkswagen, BMW, Renault, Toyota, GM, Honda, Hyundai and others linked to the mobile manufacturer sector are Sony, HTC, LG, Nokia, and Samsung (Car Connectivity Consortium, 2016). MirrorLink capabilities are supported by Android and Windows Mobile operative systems, nevertheless not all devices with these OS include the MirrorLink connectivity feature. The number of compatible devices continues to growth as new applications are certified and mobile manufacturers became aware of the market opportunity.

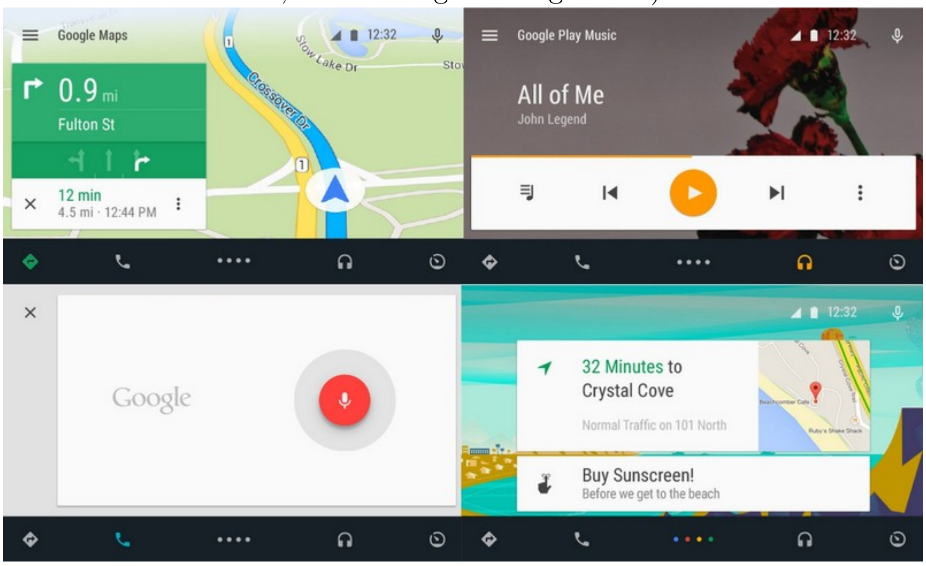
Since not every application should be used while driving, MirrorLink applies some restrictions when the vehicle is moving at a speed higher than 5km/h. In these cases, only certified application are replicated into the Head Unit screen. Applications that are intended to be used while driving must be approved and certified by the trademark Limited Liability Company (LLC), this procedure is oriented to prevent that applications that may compromise road safety be used while driving.

MirrorLink technology is very simple from the user perspective; the smartphone is connected to the infotainment system through a USB cable and after a couple of touches, the set of available applications are shown in the Head Unit screen. In the MirrorLink context, smartphones have the role of servers and Head Units are understood as clients. A server is responsible of running applications and sending user interface representations (image and audio) to the client, while a client provide input and output mechanisms to the user as speakers, microphones, display, keyboards, and others.

Android Auto is a platform promoted by Google and the Open Automotive Alliance oriented to integrate services and applications in the car using an Android Operative System Platform; in simple terms, Android for your vehicle. This standard was announced in the Google I/O 2014 and its development is based in the SDK Android L (v 5.0), therefore any Android device with a previous version is not supported. In a similar approach followed by MirrorLink, the platform is implemented mostly in the smartphone, not embedded into the infotainment system. Several auto-makers brands have provided support to this standard (Google, 2016).

Once an Android device is connected to the vehicle trough a USB cable, the systems allows to manage certain functionalities installed in the smartphone trough the Head Unit interface. Some services provided are Google Maps Navigation, music player (both streaming and offline), phone, voice interaction, and messaging. Android Auto also makes possible to interact with applications using steering wheel buttons and also monitor some internal variables of the vehicle depending on the car manufacturer agreements. Figure 2.2 shows the look and feel interface of Android Auto in 4 different contexts (navigation, music player, voice commands and Google Now service).

Figure 2.2: Android Auto interfaces (Top-Left: Navigation, Top-Right: Music Player, Bottom-Left: Voice Command, Bottom-Right: Google Now).



Apple made a similar proposal introducing CarPlay, in this case oriented to integrate IOS devices to the infotainment system of the vehicle. The mechanism allows controlling certain functionalities and applications provided by Apple or downloaded from the PlayStore using the Head Unit interface. The platform was presented in the Worldwide Developers Conference (WWDC) in 2013 and it is supported since IOS 7. CarPlay allows the management of calls, music messages, maps and voice interaction with certain functionalities trough **Siri** (Apple, 2016).

The connectivity mechanism implemented in CarPlay is similar to the used by Android, in this case the iPhone is connected to the infotainment system trough a **Lightning** cable. Once plugged in, an adapted interface is shown in the Head Unit and knobs or controls of the vehicles can be used to interact with the system. Similarly to other approaches, the main processing is performed by the smartphone. Apple also permits the development and launching of third party applications once they are certified as safe to be used on driving scenarios. In this regards, developers can create their own applications and increase the amount of available functionalities in the vehicle. Figure 2.3 shows the look and feel interface of CarPlay from the main menu.

Figure 2.3: CarPlay main menu interface (running in a SEAT Head Unit).

Besides the different approaches available for integrating smartphones functionalities to vehicles, vehicle manufacturers have also proposed new ways for interacting with these functionalities. The majority are oriented to guarantee natural voice interaction with the system, under this approach the user does not have to remove the hands of the steering wheel or deviate the eyes off the road. Most of the brands already provide voice interaction services, nevertheless the technology is not very mature to be used widely.

Other alternative for interacting with applications is the implementation of control by gesture. These gestures could be recognized trough touchable interfaces, e.g., the user drag a pattern in the screen, or by special cameras, e.g., air gestures close to a sensor located in fixed place. Several brands are starting to integrate recognition systems for controlling simple features as volume control and call answering. The technology is not yet very mature and depending on the gesture, requested commands are not always recognized properly.

Vehicle manufacturers have also added interfaces located in several places within the vehicle in order to ease the visualization of information and the user interaction with functionalities. Examples include steering wheel buttons with the goal of prevent the driver releases the steering wheel, Head Up Displays (HUD) aimed at reducing the glance time of drivers and different control interfaces such as rotatory encoders or touchable interfaces that are easy to handle. Some of these interfaces can have positive effects by reducing the driver concentration required to interact with services, while others may have the contrary effect and increase distraction. As a result, the design of infotainment systems must be careful planned in order to ensure both safety and driver experience.

2.1.2 HMI design principles

The design of a proper HMI interface for the purpose of ensuring a good user experience and an adequate road safety has been extensively studied; as a result, several guidelines and principles have been put forward. This section provides some background on the principles most widely used by developers and designers when defining an interface. These concepts are relevant for our purpose, since their baseline will be used to propose adjusted principles for adaptive interfaces in vehicles. The majority of these principles are based on accumulative experience and heuristic experiments obtained through user feedbacks. Therefore, the results are closely related to human nature, to the way the mind processes information, as well to cultural and physiological factors.

As regards road safety, the European Statement of Principles on human machine interface (ESoP) has proposed a series of recommendations in the areas of design, installation, information presentation, and system behaviour of in-vehicle information and communication systems (Commission of the European Communities, 2008). From this set of recommendations, those with most relevance to HMI design are as follows:

- The system should not give rise to potentially hazardous behaviours and should prevent the distraction of the driver.
- The information should be displayed in a uniform manner and be consistent among available interfaces.
- Relevant information should be clearly displayed and the user should be able to detect and understand it with a few glances.
- Priority should be assigned to the information so that may be displayed as rapidly as possible. Furthermore, it should be organized in such a way as to avoid the presentation of simultaneous messages and be displayed in a sufficient amount of time. Similarly, auditory messages should not mask high priority messages or warnings.
- The manual-visual action function should not be interrupted, although exceptionally it may be cancelled automatically or by the user when a certain amount of time has passed; for example, the driver should be able to enter half of a phone number, continue driving for some time and then go back to finish writing the phone number.
- It is recommended that information unrelated to primary driving tasks should not be displayed while the vehicle is in motion. Moreover, the information should not be restored immediately when the vehicle stops, but only after a certain amount of time.

As regards the improvement of user experience and usability of the system, two notable research works in the area are: the *Eight Golden Rules* for design interaction proposed by Schneiderman (Shneiderman, 2010), and the *10 Heuristic usability principles* defined by Jakob Nielsen as an extension of the Golden Rules (Nielsen, 1993).

Schneiderman proposed the set of 8 rules derived from experience and heuristic experiments; these rules are recommended in order to achieve a well-designed interface. The most significant design concepts contained therein are: consistency in the design style; always offer informative feedback about system status (loading, successful result, errors); provide easy reversal of actions to allow the user to undo or cancel started actions, and finally reduce short-term memory load so that the user is not obliged to remember information from one display or another.

In a similar research work, Nielsen performed experiments in order to detect usability problems by means of heuristic tests. In his results, he defined 10 basic principles, known as "heuristics", which can be used as guidelines for developing user interfaces. These principles were chiefly devoted to web design, but as HMI infotainment systems are quickly evolving and becoming increasingly more complex, this background provides a useful guide to design.

Some of these principles are similar to the Schneiderman's Golden Rules, the most relevant of which are as follows: present an indicator about the system status in order to provide an understanding about what is happening with the system; the user should easily be able to cancel operations or return to previous views in case of mistakes or changes of opinion; the design must be consistent as regards colours, words and icons in the application; the interface should display only the essential information in order to avoid overcrowded screens that may lead to confusion, and finally, shortcuts should be accessible for executing the most common actions.

In another research, Gruhn sets out a set of concepts that may also be regarded as a design reference. He differentiates four main notions that should be taken into account in every design: Contrast of relevant things among the rest (emphasis); repetition of visual elements, alignment and interconnection between them, and proximity differentiation of linked or unrelated elements (Gruhn, 2011). This research also points out the relevant importance of so-called *Situation Awareness* (SA), by which the user must always be conscious of the purpose of the current interface; how to access other contexts and have an overall idea about what will happen when an interaction is started.

2.1.3 Summary

The first part of this section summarizes the history of infotainment evolution, this introduction is oriented to forecast what will be the trend of such systems during the coming years. It is expected that infotainment system will continue to integrate more interfaces, adding larger screens and replacing buttons with touchable panels. New services and functionalities are also be added which will increase the driver workload due to greater interactions with these systems. Given this premise, a proper management of the information regarding user experience and road safety must be ensured.

The section also presents some of the existing design principles for creating in-vehicle infotainment systems. These principles are widely taken into account in the design of infotainment systems by vehicle manufacturers. However, few details are given with regard to the integration of new interfaces such as HUD, HMD and gesture sensors to cite some examples.

2.2 The Activity of Driving

According to several statistical reports published by European organizations, vehicles are the most common transportation mode in the EU. The flow of passenger in cars, measured in passengers per kilometre (pkm), represents 73.7% of total personal transportation flow in the EU, 79.5% in the USA and 41% in China as reported by the European Commission in 2015 (European Commission, 2015b). An analysis of post-harmonized National Travel Surveys (NTS) of different European countries arrived at the same conclusion (Christensen, 2013), their report shows that the average amount of time travelled in vehicles represents the 57% of total travelled time and about the 60% of total travelled distance per person among all transportation modes.

Unfortunately, vehicles are also the transportation mode with more fatalities registered per kilometre travelled as reported by (European Transport Accident Statistics Working Party, 2003), where is displayed that 97% of all transport deaths in the EU are due to road crashes. In 2015, 26,300 deaths were registered in the EU due to driving accidents, while in the USA 32,675 fatalities were recorded for the same reason in 2014 (Safety and Index, 2016) and (National Center for Statistics and Analysis, 2016).

A comparison between the number of deaths in different transportation modes is presented in Table 2.1. The data reflect a higher number of deaths per kilometre and hour due to road accidents than due to any other transportation mode. With respect to the distance travelled, the amount of deaths per kilometre due to road accidents in cars is the double than to any other of non-road transportation mode. In terms of travelled time, motorcycles clearly represent the higher risk of death, followed by foot travels, and continuously with equal risk cycles and cars.

Table 2.1: Deaths per distance and time for different transportation modes in the EU.

Transportation mode	**Deaths per 100 million person - kilometres**		**Deaths per 100 million person - travel hours**	
Road (Total)		0.95		28
Motorcycle	13.8		440	
Foot	6.4		75	
Cycle	5.4		25	
Car	0.7		25	
Bus and coach	0.07		2	
Ferry		0.25		16
Air (civil aviation)		0.035		8
Rail		0.035		2

Source: (European Transport Accident Statistics Working Party, 2003).

The causes of driving accidents have been extensively studied and can be classified into two main groups: human causations and vehicle/infrastructure causations. According to the International Road Transport Union (IRTU), human errors is the leading cause of accidents which accounts for up to 85.2% of the total number of accidents recorded (Copsey et al., 2011). These errors can be attributed to: (1) exogenous factors that are associated with the complexity of the surrounding environment and (2) endogenous factors given by the driver profile, driving experience and driving behaviour (Egeth and Yantis, 1997) and (Engström et al., 2013).

In the following subsections, the most relevant exogenous and endogenous factors that have been shown to have a higher impact on road safety are presented. The influence level of each factor is difficult to determine since accidents are random events in which a big amount parameters are dynamically changing the driving scenario. Most of the research in safety calculates the risk of accidents based on empirical data and experience, therefore, most the arguments and statements are based on statistical data.

2.2.1 Exogenous factor that affect driving activity

The performance in the execution of driving tasks is highly dependent on the complexity of the surrounding scenario; roads are heterogeneous and may be urban, rural, freeway-type, roads with bends, have different traffic density, to say nothing of changes in the environment such as weather or the presence of animals, pedestrians and cyclists (European Commission, 2015a) and (Teh et al., 2014). The influence of all these parameters that affect driver performance have been deeply studied, and the most common method for assessing risk of crash is through the comparison of statistical information.

The European Commission (European Commission, 2015a) and the National Highway Traffic Safety Administration (NHTSA) (NHTSA, 2014) include in their reports a comparison of accidents depending on parameters that characterize the driving scenario. A list of exogenous factors considered to have the greater influence on driving complexity are presented below.

Road type

Most of road statistical reports highlight the influence of road type and road area on crash risk. The Annual Accident Report of the EU (European Commission, 2015a) classified the amount of accidents in terms of the road area, road type and junction type where they occurred. Regarding the road area, it is found that 38% of accidents occurred inside urban areas, a 8% on motorways and a 54% on non-motorway roads. This result shows a lower risk of collision on the motorways compared to the rest, which does not necessarily indicate a lower severity in these accidents.

The impact of road type on attention level was deeply studied in a project named Human Machine Interface And the Safety of Traffic in Europe (HASTE) (Östlund et al., 2004). The research evaluated the effect of three different road types (urban, rural and motorway) on visual glance patterns using an eye tracking device, cognitive load using self-measurements and driving performance using vehicle variable indicators. Findings shown similar results for most of the metrics with a slight difference in terms of cognitive loads for each scenario.

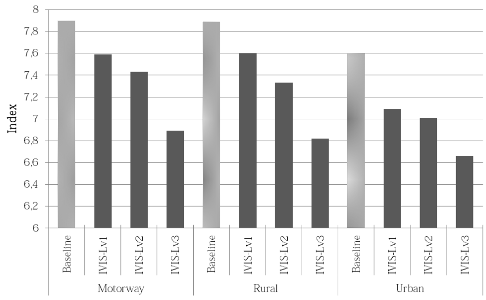
Figure 2.4 presents four charts where the effects of three road areas (Motorway, Urban and Rural) on four different metrics (Self-report, mean speed, hear rate variability and reaction times) are compared. As an additional comparison metric, three different IVIs with diverse complexity interaction levels are evaluated (IVIS-Lv1 is the easiest to use, and IVIS-Lv2 is the one with a higher complexity associated in terms of interaction).

Regarding self-report measurements (Figure 2.5a, where 1 represents the opinion of driving extremely poorly and 10 the thought of driving very well, motorways are reported as the road type driven with a higher quality, followed by rural roads and finally urban areas. Figure 2.5b shows the mean speed followed when driving on different types of roads and interacting with different IVIs, in this case small different are seen in when comparing different IVIs for same road types, but a clear difference associated to average speed deepening on the road type is observed.

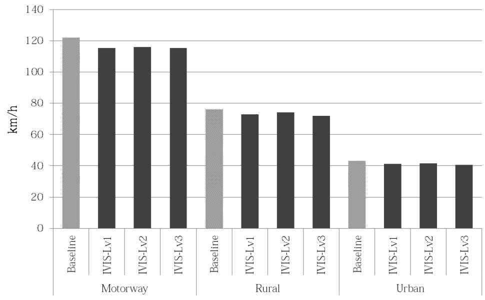
The study also measured the heart rate variability in terms of the average 0.1Hz component of Heart Rate Variability (HVR) as is presented in Figure 2.5c. In this case, small differences are perceived, but it seems that urban areas are the one with higher variabilities in the heart rate. Finally, Figure 2.5d shows the reaction time of drivers in different road areas; urban areas are associated to the higher reaction times. This result may be due to the fact that highways are less variable in topology and complexity when the traffic level is low compared to other types of roads where the scenario is usually more dynamic (curves and speed variations).

Similarly, in the HASTE project was studied the impact of the road topology on driving workload. The study concluded that roads with curves have a higher effect on the increment of driver workload in comparison to straight roads. Analysis were made in terms of self-reported driving performance, percentage of correct responses to secondary tasks, reaction times and driving performance metrics as lateral and longitudinal control.

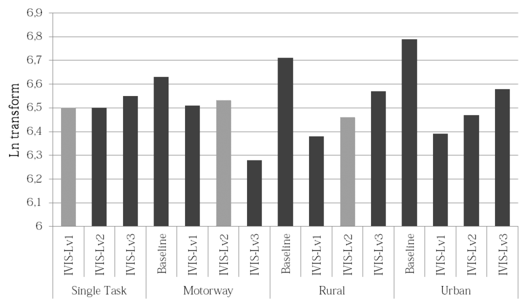
Figure 2.4: Effects of road area and road type based on several metrics.



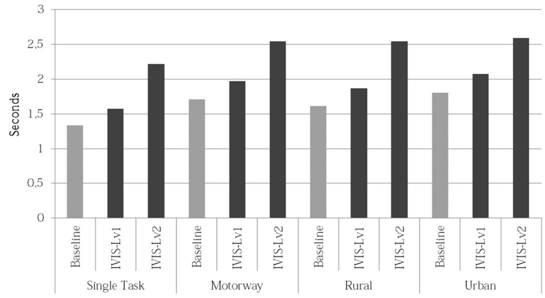
(a) Self-Reported driving quality –index–



(b) Average Speed –km/h–



(c) Average 0.10Hz component of Heart Rate Variability (HRV) after a natural logaritmic transformation –Ln-transform–



(d) Mean reaction time –seconds–

Source: HASTE Deliverable 2. (Östlund et al., 2004).

Regarding the type of junctions[1], The Annual Accident Report of the EU (European Commission, 2015a) shows that the majority of accidents occurred in non-junction road (80%), while a 7.9% in a crossroad[2], a 5.1% in a T or Y junction[3] and a 7% among the rest of types. The NHTSA (Highway and Safety, 2008) presents a comparison of the case vehicles involved in road accidents depending the number of lanes; in the report about 52% were involved in crashed in roadways with more than three lanes, 45.5% in two lane roads, and 2.6% in single lane flows.

Some analysis also provide information about the vehicle movement prior to the crash. According to the NHTSA report, in 46.6% of crashes the previous manoeuvre was going straight, in 21% was negotiating a curve, a 16% was stopped in a traffic light and a 6% decelerating in a traffic line. This results confirm that the crash risk is influenced by more variables than the road topology, since straight roads are the lower demanding road types and nevertheless has the majority of accident occurrences.

Traffic level

Regarding traffic levels, several studies have shown its impact on driver performance. De Waard studied the driver behaviour and its driving performance in difference traffic conditions, where traffic density and presence of Heavy Good Vehicle (HGV) were used as metrics

[1]Road intersection with three or more arms.
[2]Road intersection with four arms.
[3]Road intersection with three arms.

(De Waard et al., 2008). He found a higher variability in drivers travel speed when traffic density was increased. In a similar way, the minimum time headway [4] was negatively correlated to the proportion of HGVs in the road. Joining motorway traffic was also considered to involve greater effort and crash risk when HGVs were present.

Teh evaluated how variations in the traffic density affect the driver subjective workload and driving performance (Teh et al., 2014). The study compared driving performance metrics and self-assessment reports of drivers who were asked to drive in a driving simulator. Traffic levels where predefined to three levels according to the density of vehicle in the road: low, medium and high. In terms of subjective measures, results shown that segments with higher levels of traffic were associated with higher levels of perceived workload. The driving performance measures demonstrated changes in longitudinal and lateral control depending on the traffic density, these effects were linearly correlated with traffic density.

Environmental conditions

The impact of weather on driving behaviour and performance has also been deeply studied in many researches. Empirical evidence indicating that adverse weather conditions are associated to a higher number of accidents in roads has been exposed in (Andrey et al., 2001). Collision risk increases in about 50% during precipitations and variations in the number of occurrences have been found depending on the intensity and form of the precipitation, e.g., snowfalls have a greater effect than rainfall on collision occurrences but crashes use to be less serious.

A statistical report of the Institute for Road Safety Research also highlighted the influence of weather on road safety (Institute for Road Safety Research, 2012). During showers, drivers change their driving behaviour reducing speed and increasing the distance to the next vehicle. Nevertheless, these countermeasures are considered insufficient and the risk of being involved in a crash is still higher than during dry weather. The problem of precipitations is associated to reduced visibility and low friction of the road surface. The conditions caused by the formation of ice are even more hazardous given a higher reduced adhesion of the road surface.

Sunrises and sunsets are also reported to have negative effects on driving performance, a bright light close to the horizon have a strong impact on visibility; glares are attributed to be responsible of more crash accidents than adverse weather conditions according to (Highway and Safety, 2008). Regarding the temperature, little research has been found but an increment in crash accidents has been reported during heat waves. High temperatures have physiological and psychological effects on some drivers: their reaction times increases, they become more irritable, get tired and loose concentration (as
cited in (Institute for Road Safety Research, 2012).

The presence of pedestrian, animals, cyclists or any other obstructions in the road also increases the driving complexity since more variables must be processed and considering when performing manoeuvres. Little research exists in this respect, but it is expected that the addition of more stimulus to the driver will cause a higher attentional demand and therefore a reduction in the performing of driving tasks adequately.

[4] "difference between the time when the front of a vehicle arrives at a point on the highway and the time the front of the next vehicle arrives at the same point". Source: Traffic Flow Theory, (Gartner et al., 1997).

2.2.2 Endogenous factors that affect driving activity

Endogenous factors, on the contrary to exogenous factor which are given by environmental conditions, can be directly linked to an incorrect execution of the driving task. In a report presented by the NHTSA in 2008 crash accidents attributed to drivers were classified in four main groups: recognition errors, decision errors, performance errors, and non-performance errors (Highway and Safety, 2008).

Table 2.2 shows the percentage of crashed imputed to each driver error and specific cause. The greater amount of crashes are associated to recognition errors (about a 40.6%) and decision errors (34.1%), the rest of error types represents a 25.3% of the total. The table shows how recognition errors, mainly associated to distractions, lack of proper attention or overload of driver capacities have a clear implications on road safety. On the contrary, performance errors such as overcompensation, poor direction control have a lower relation to crash events.

Results shown in the table are useful for understanding how important it is for drivers to concentrate on the task of driving, avoid distractions and do not assume risky behaviour while driving. Not only the own driver' safety may be compromised, but the whole road scenario could be affected by an wrong manoeuvre of one single driver.

Table 2.2: Pre-crash event attributed to drivers.

Critical Reason for Critical Pre-Crash Event		Weighted Percentage of number of crashes
Recognition error	Inadequate surveillance	20.3%
	Internal distraction	10.7%
	External distraction	3.8%
	Inattention (i.e., daydreaming, etc.)	3.2%
	Other/unknown recognition error	2.5%
	Subtotal	40.6%
Decision error	Too fast for conditions	8.4%
	Too fast for curve	4.9%
	False assumption of other's action	4.5%
	Illegal manoeuvre	3.8%
	Misjudgement of gap or other's speed	3.2%
	Following too closely	1.5%
	Aggressive driving behaviour	1.5%
	Other/unknown decision error	6.2%
	Subtotal	34.1%
Performance error	Overcompensation	4.9%
	Poor directional control	4.7%
	Other/unknown performance error	0.4%
	Panic/freezing	0.3%
	Subtotal	10.3%
Non-performance error	Sleep, actually asleep	3.2%
	Heart attack or other physical impairment	2.4%
	Other/unknown critical non-performance	1.6%
	Subtotal	7.1%
Other/unknown driver error	Subtotal	7.9%
Total		100.0%

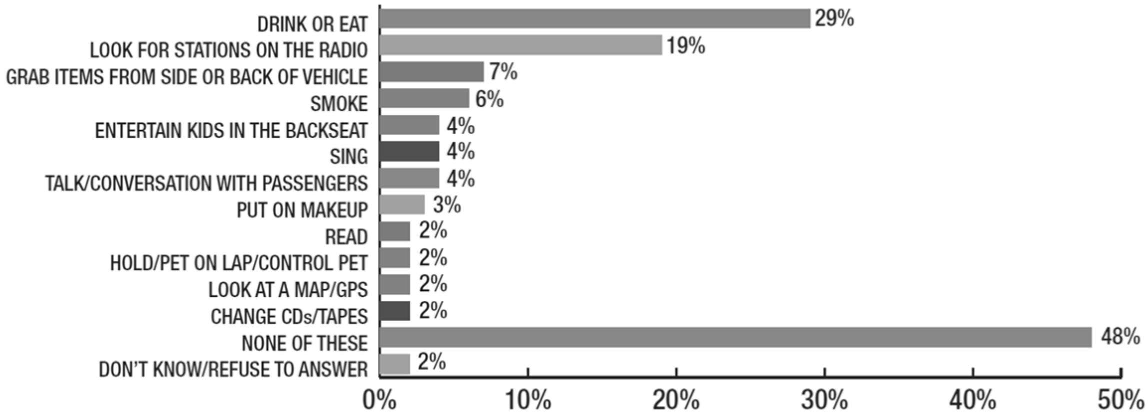
Source: NMVCCS, (Highway and Safety, 2008).

The activity of driving is complex and small distractions or errors have major consequences. In 1970, McKnight and Adams suggested that drivers must learn approximately 40 primary driving in order to perform the driving activity successfully (Shinar et al., 1998).

The additional introduction of secondary tasks as talking, eating or interacting with the radio interferes with these primary tasks affecting driving performance and compromising passengers safety. According to a report by the U.S. Department of Transportation, 10% of fatal crashes and 18% of injury crashes in 2014 were reported as distraction-affected crashes (National Center for Statistics and Analysis, 2015).

The effects of distraction on crash accident risk were also evaluated by Swanson, identifying the great consequences of performing secondary tasks while driving on road safety (Swanson, 2010). His studies exposed that 38% of respondents claimed to have been hit or nearly hit as a result of other driver being distracted by the usage of technologies while driving. Figure 2.5 shows the percentage of secondary tasks performed while driving in 2010 (self-reported by drivers). Drinking or eating are the most common tasks representing a 29% of the total, while the sum of tasks related to drivers' interaction with technology rises to 25%; it is expected that this percentage had increased in recent years due to the increment of services available.

Figure 2.5: Secondary tasks conducted while driving.



Source: Harris Interactive (Swanson, 2010).

The "Real Automóvil Club de Cataluña" (RACC) classify the distractions in four main groups; visual distraction when the driver is not looking the road due to the execution of a secondary task, auditive distraction when the driver is focusing the attention in a sound or a voice, biomechanical distraction when the driver is interacting with an object not mandatory for performing driving tasks, and cognitive when toughs are abstracting the driver and interfering with the driver tasks (Pérez et al., 2008).

The same study exposed that road safety implications of distractions are positive correlated with three parameters: complexity, duration and frequency of the tasks. Table 2.3 shows a classification of secondary activities according to these parameters. The study shows that even when a danger ranking of distractions has not been established, complex activities have the greater impact on road safety independently of their duration and frequency. Dangerous activities are also characterized by high durations and frequencies regardless of the complexity level, this occurs as a result of dynamic driving scenarios where hazardous events occur in a random manner most of the time.

The skills and ability required to execute driving-related tasks vary from person to person; they consist of a series of cognitive and physical features that characterize each individual. Significant findings have been reported regarding the impact of profiles features as age (Road, 1992), driving experience (Deery, 2013) and (Lancaster and Ward, 2002),

Table 2.3: Classification of secondary activities in terms of Complexity, Duration and Frequency.

Activity	**Complexity**	**Duration**	**Frequency**
Try to reach an object	High	Low/Medium	High
Locate and answer a mobile-phone	High	Low/Medium	Medium
Dial a number in a mobile-phone	High	Low	Medium
Read a text or a SMS	High	Medium	Medium
Set a destination in the navigation system	High	Medium	Medium/Low
Attend children or animals in the back seat	High	Medium	High (if apply)
Put on make-up	High	Medium	Medium
Look objects outside the vehicle	Medium	Low/Medium	High
Insert/Remove a CD	Medium	Low	High
Talking on the mobile-phone	Medium	High	Medium
Eat/Drink	Medium	High	Medium/High
Smoke	Low	High	High (if apply)
Have a conversation with a passenger	Low	High	High
Climate/Windows controls	Low	Low	High
Turn on/off the radio	Low	Low	High

Source: RACC, (Pérez et al., 2008).

driving style (Elander et al., 1993) and gender (Lancaster and Ward, 2002) on road safety. Likewise, some studies have demonstrated a close relation between different profiles and risk perception, which is linked to the way in which drivers face different scenarios (driving attitude) (Nilsson, 2004).

These parameters are not universal indicators of crash risk since each individual is different and cannot be uniquely characterized in terms of few profile variables. Nevertheless, statistical data have shown certain patterns that allows to make some inferences regarding which profiles are more susceptible to be involved in accidents. Moreover, these parameters are usually taken as reference by assurances to regulate their price policy.

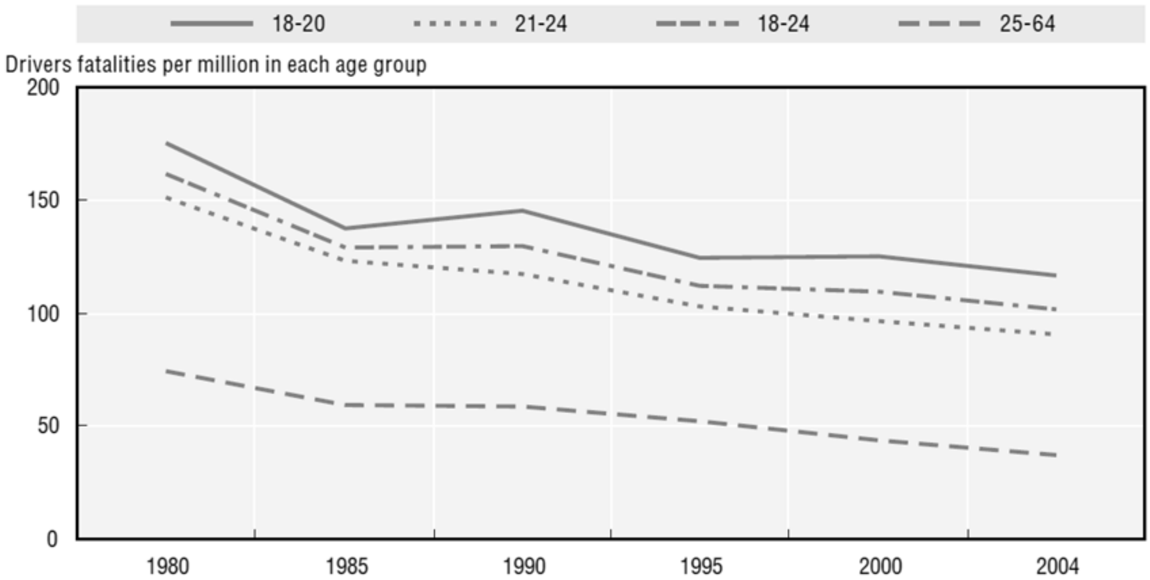
Age

Statically, young drivers are more susceptible to be involved in crash situations than older drivers. People between 19 and 36 years old are twice likely to have a crash accident than older drivers with ages between 40 to 55 years old (Lancaster and Ward, 2002). Similar results were reported in (Co-operation, 2007) where four age groups were compared in terms of fatalities due to crash accidents. Figure 2.6 shows information in this regards; it is seen how the group age between 25 to 64 years old presents a significant lower number of fatalities among all.

The higher amount of accidents for younger drivers have been attributed to inexperience, physical maturity and lifestyles; many accidents were also reported as consequence of speeding or driving under the influence of alcohol or drugs. In general, younger drivers are more likely to drive at faster speeds and be involved in more driving violations such as being under the influence of alcohol or drugs than older drivers (Lancaster and Ward, 2002).

It has been also highlighted that the main reason that young and novice drivers get

Figure 2.6: Driver fatalities comparison in terms of drivers age. (Austria, Great Britain, the Netherlands, Sweden, Switzerland, US).



Source: IRTAD (Co-operation, 2007).

involved in more accidents is their low driving experience (Road, 1992). When learning to drive, primary related-tasks as changing gears, looking in the rear-view mirror, steering, braking at time, and taking route decisions easily overload the driver capacity and errors are more probable to happen. Additionally, younger drivers are more likely to adopt riskier driving styles as a consequence of a lower physical and emotional maturity. Young people experience an intense social life which may leads them to show off and be more susceptible to peer pressure (Road, 1992).

Gender

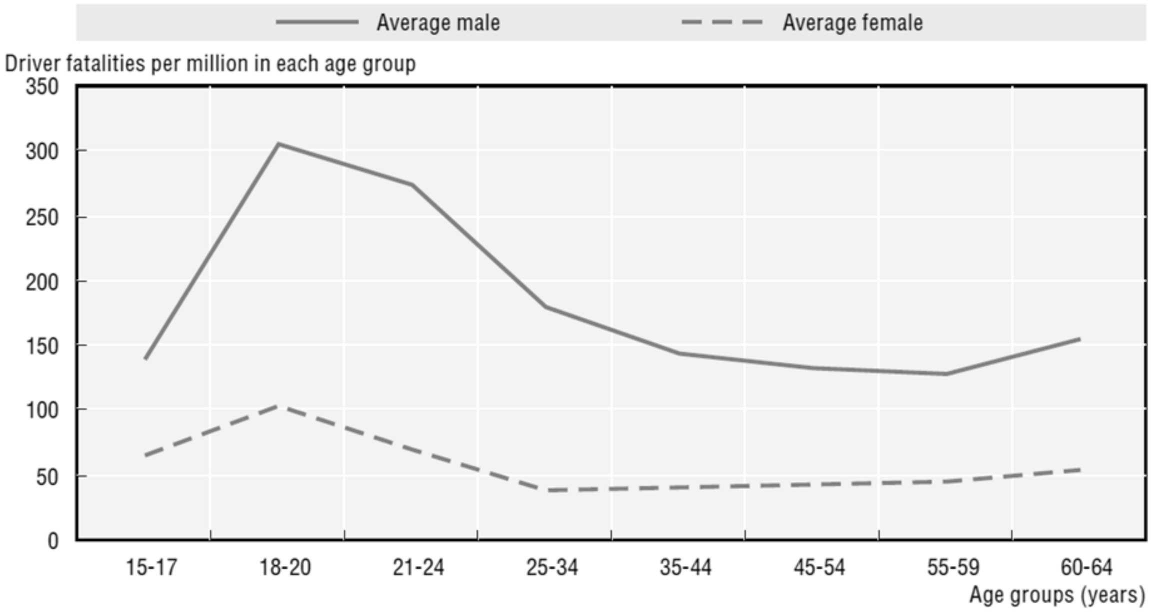
Different studies have reported that men are involved in more crashes than women and that these tend to be more severe. Figure 2.7 shows a trend comparing the average number of fatalities per gender for different age groups; it is seen how the number of fatalities follow a similar pattern for both gender in terms of the age. However, men are involved in more fatal crashes than females for every age group, usually more than triple.

It is noticed that even when young males drive more than young women and therefore should have more driving experience, they are anyway involved in more crashes than women per kilometre (Co-operation, 2007). This may be explained by a more hazardous behaviour and riskier driving style adopted by men in comparison to women; in general, women seem to be more concerned about safety-related issues (Dodd and Mills, 1985).

Also, men are more likely to overestimate their driving skills and avoid compensations manoeuvres according to the driving scenario complexity. Men are generally involved in accidents resulting from speed; whereas women appear to be involved in more accidents as a result of perceptual and judgemental errors (Lancaster and Ward, 2002).

Another factor to take into account is the driving scenario in which different gender usually drive, young females commonly drive in built-up areas where crashes are likely to be less severe, while young men tend to drive during leisure time and at night. Moreover,

Figure 2.7: Driver fatalities comparison in terms of drivers age and gender. (Austria, Great Britain, the Netherlands, Sweden, Switzerland, US).



Source: IRTAD (Co-operation, 2007).

the majority of reports consistently show that males are more likely than females to report driving under the effect of alcohol (Lancaster and Ward, 2002).

Driving experience

Driving experience parameter is also very associated to crash risk; several researches have shown that the risk of being involved in a crash during the first year of driving considerably decreases for the following years (Road, 1992). Figure 2.8 shows a graphical comparison of the number of crashes depending on the age they obtained their driving license (Lancaster and Ward, 2002).

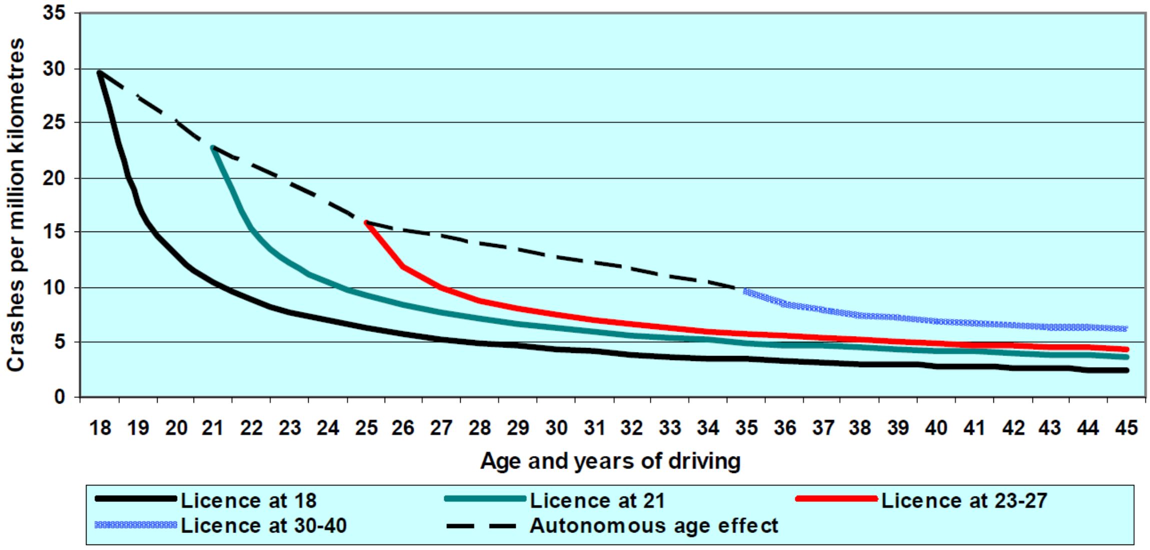
It is seen how people with licenses obtained at a more advanced age have a higher risk of being involved in a crash than those who obtained their license earlier. This pattern is maintained during all the driving years of a person, therefore in average, a person who obtained the driving license at 25 years old will have more chances of being involved in an accident than other with the same age that obtained the driving license at 18 years old.

The benefits of driving experience is also higher when the license was obtained at an early age. A strong indicator of the importance of driving experience is observed in the last part of the graph (Age and years of driving higher than 40). Here, people who obtained licenses before 22 years old have the lowest reported of crashes among the rest.

Personality

In a similar manner as gender and age are linked to risk perception, personality is the probably better indicator about each individual behaviours, preferences and expectations. Different approaches for evaluating personalities have been proposed; one of the more widely accepted is the "Five Factor Model" (FFM) in which the structure of personality can be

Figure 2.8: Driver fatalities comparison in terms of years of driving. (Austria, Great Britain, the Netherlands, Sweden, Switzerland, US).



Source: SWOV (Vlakveld, 2004).

represented in terms of: Extroversion, Neuroticism, Conscientiousness, Agreeableness and Openness (Road, 1992).

The attribution of road accidents to individuals with certain types of personality has been discouraged by several studied since there is not empirical evidence that support this hypothesis. Weak relations in this regard are attributed to the randomness nature of crash accidents as well to other factors in the environment that may have a higher influence.

Even so, some researches have found interesting patterns that link Sensation - Seeking (SS) –defined as the need to experience novelty, ex
citepppment and danger– with crash involvement in traffic. Drivers with the highest values of SS demonstrated to be more likely to engage risky driving behaviours such as speeding, traffic violations and faster steering manoeuvres.

Aggressive drivers also tend to be involved in more crash accidents, this factor is measured comparing those drivers who have a lower capacity of controlling hostility and anger. The level of aggression is usually measured using the "Buss-Durkee Hostility Inventory" scale, which has been positive correlated with an increment in crash events and traffic violation events of drivers (Beirness, 1993).

Physiology

Physical health and impairments have also demonstrated to have an influence on driving performance (Lancaster and Ward, 2002). Driving regulations usually add as requirements for acquiring a driving license a medical certificate that must be granted by a medical institution in order to endorse sufficient healthiness such as proper vision and good motor-manual control.

Stress has also been considered a likely cause of increment in risks of accidents. Several researches found that drivers with higher levels of stress, feeling rushed, and lower life satisfaction scores were involved in more accidents that the rest (Lancaster and Ward, 2002).

Fatigue, understood as being tired or sick, is another factor quite related to driving performance (Hs, 2013). From the total amount of crashes, a 7% of drivers reported being fatigued. This may be a consequence of quantity and quality of sleep, large or monotonous driving scenarios, low driving experience, illness and being under the influence of alcohol, drugs or medications. Green found that a 0.05 Blood Alcohol Concentration has the same effect on driving performance that staying awake for 18 hours (Green, 2000).

Speeding

Speeding is frequently associated to crashes on roads; about 10% of all traffic accidents in the USA were imputed to violating speed limits (as cited in (European Commission and Energy, 2009)). This risk factor can be expressed as a function of some profile parameters such as drivers' age, gender and personality. Taylor stated that each individual have a different subjective appreciations about going too fast or going too slow, these appreciations depends on exogenous factors such as road environment, traffic level and weather conditions (Taylor et al., 2000).

The relationship between speed and accident risk has been deeply studied by several authors, Nilsson proposed a model based on empirical data where the number of injury accident rates is given as a function of the speed (Nilsson, 2004). In his model, expressed by Equation 2.1, the number of injury accidents A_2, is given by the previous number of accidents A_1 and the relation of previous speed v_1 and final speed v_2 to square of two. The same formula can be used for getting information about the severity of the crashes when is used the power of 3 and information about fatal crashes using the power of 4.

$$A_2 = A_1 \left(\frac{v_2}{v_1} \right)^2 \tag{2.1}$$

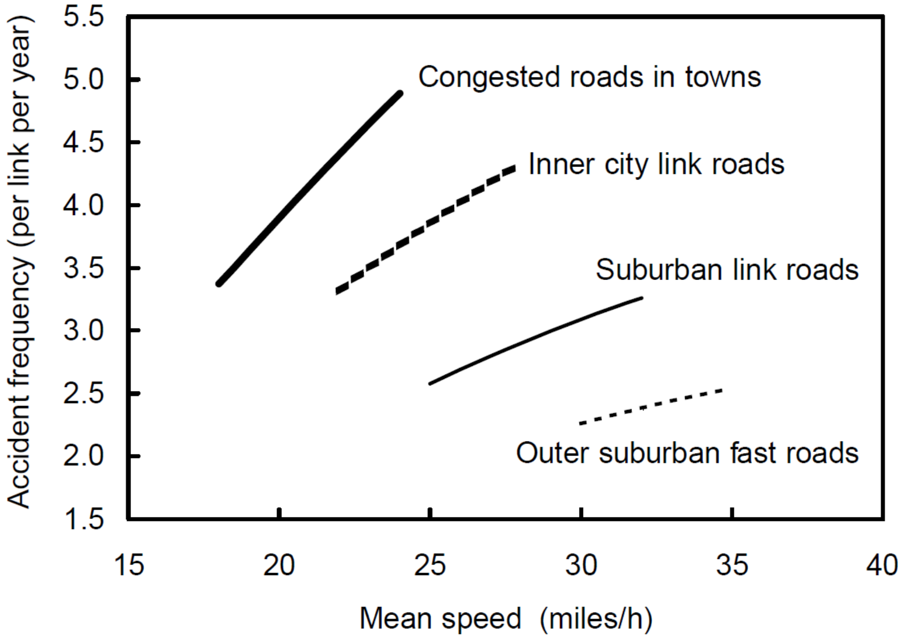
According to Nilsson's model, which has been validated in several studies, a change in average speed of 1 km/h represents a change in the number of accident between 2% for a 120 km/h road and 4% for a 50 km/h road. Further studies evaluated Nilsson' model in different road types finding a higher effect of an increase or decrease in a rural road than in a urban road.

Figure 2.9 shows a graphical representation of the impact of speed on accident frequency, as well as the variability of results depending on the road type (Taylor et al., 2000). It is seen how the slope of the different curves have a similar behaviour, nevertheless the effect of speed in *congested roads in towns* is much higher than any other type of road. On the contrary, suburban roads are the less affected in terms of an increment in the mean speed of travel.

Situation Awareness (SA)

An important concept commonly exposed in literature regarding driving activity is Situation Awareness. The term can be simply expressed as 'knowing what is going on' (Walker et al., 2001). An individual with a proper level of SA will be capable of detecting relevant variables in the environment with respect to time and space, comprehend their meaning and take optimal decision based on expected changes in a near future. From a driver perspective, the key components would be knowing the vehicle position, the relative position of other vehicles and the road topology. Not only at present time, but also the perception of expected behaviours and changes must be achieved. Several researches have shown that an inadequate level of SA is one of the primary causes of accidents.

Figure 2.9: Accident frequency vs mean speed fro different road types.



Source: Transport Research Laboratory (Taylor et al., 2000).

Three methods are used by Stanton for measuring the level of SA in naturalistic and simulated environment: concurrent verbal transcriptions, probe recall and self-report methods (Stanton et al., 2015). Results exposed the relevance of giving proper feedbacks to the user in order to increase SA while driving.

2.2.3 Workload and driving complexity

Driving is a complex and dynamic activity in which the attention level required is constantly changing. In 1970, McKnight and Adams suggested that there were approximately 40 primary driving tasks and 1,500 related sub-tasks that a driver must learn in order to perform the driving activity successfully (McKnight and Adams, 1970). A complete "Hierarchical Task Analysis of Driving" (HTAoD) where the driving activity was represented as a set of programmable instructions shown that there were needed 1600 bottom-level tasks and 400 plans of actions were needed to replicate the driving activity (Stanton et al., 2015).

In order to understand the driving activity and its complexity, it is mandatory to understand how the mind collects the variables from the environment. Perchonok and Pollack in 1981 (Perchonok and Pollack, 1981) described 4 processing stages. Detection, when the driver becomes aware of certain input, Identification, by which the stimulus is interpreted and understood, Decision, when the driver decides what action to take depending on the identified stimulus, and Response, the final stage when the muscles receives the instruction to carry out the planed action.

Similarly, Michon (Michon, 1985) classified the driving tasks in three levels: a strategic level related to planning the route to the destination where are analysed variables such as traffic, time to arrive, and preferable route. A second level called tactical which is related to the execution of manoeuvres to reach the destination, for example crossing intersections, overtaking cars, integrating a stream of moving vehicles, waiting at traffic lights. And finally a third level called operational that contains more basic and physical tasks such as braking, accelerating, moving the steering wheel or shifting gears. The complexity associated to these levels is represented in Table 2.4.

Table 2.4: Different task levels and complexity associated.

Tasks Levels	Complexity	Activity (Physical)	Duration
Strategic	High	Low	High
Tactical	Medium	Medium	Medium
Operational	Low	High	Low

Given the big amount of parameters which are constantly changing the driving scenario, it is not always possible for drivers to process all the information. As a consequence only a part of it is selected for interpretation, this selection process is known as Attention (Sofia and Pereira, 2009). Broadbent (Broadbent, 1952) supported the existence of a bottleneck in the information processing mechanism that prevented the simultaneous processing of multiple information channels (Selective Attention).

This bottleneck model was replaced latter by the Divided Attention Model (Moray in 1967). He proposed than concurrent activities can be attended at the same time but there is a limited capacity for attending all of them successfully. From this idea came up the single-resource theory, this concept was based on the assumption that people possess a unique resource of attention that limits the capacities of available cognitive process for performing a task.

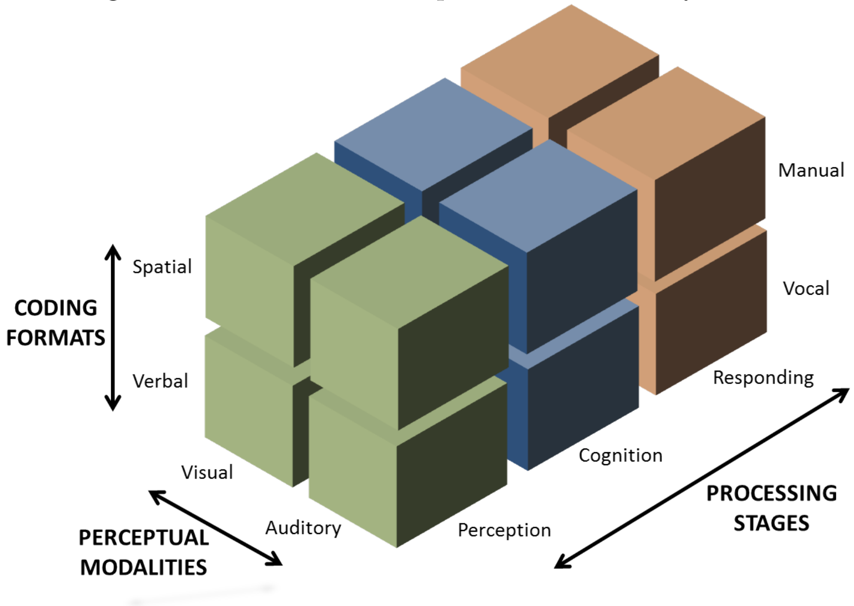
Moray proposed in 1967 a divided attention model in which concurrent activities can be attended at the same time but there is a limited capacity for attending all of them successfully. From this idea came up the single-resource theory, this concept was based on the assumption that people possess a unique resource of attention that limits the capacities of available cognitive process for performing a task.

For the following years, several authors opposed to the single-resource theory; they were based on the idea than people have several resources of attention, each one dedicated to attend a particular processing job (multiple-resource theory). Allport in 1972 demonstrated that pianists were able to read music while shadowing a speech, maintaining a good performance in both tasks (Allport et al., 1972).

Wickens defined in 1984 one of the more comprehensive models of attention in regards of multiple-resource theory (Wickens, 1984). He defined a three-dimensional model expressed by the stage of resources, the modality and the processing code as is shown in Figure 2.10. The stage of resources contains the distinct phases of the processing mechanism of information: Encoding, Central Processing and Responding. Modalities define the way inputs reach the subject and it can be Visual or Auditory. Codes define how it is translated the input depending the actual stage of resources: Spatial or Verbal for the Encoding and Central Processing Stages, or Manual or Vocal in the Responding Stage.

In Wickens's Multiple-Resource Model two different tasks interfere between them only if they go through the same channel. For example, the task of identifying a signal and expressing its meaning aloud (which traverse the Visual-Spatial Encoding, go through the Verbal Central Processing Stage and finally, is processed by the Vocal Responding) will interfere with other tasks such as detecting a pedestrian on the road and announcing his or her presence.

Figure 2.10: Wickens' Multiple Resource Theory model.



Engstrom proposed a general taxonomy definition for categorizing inattention during driving, the taxonomy is based on 12 key principles where is highlighted the adaptive nature of drivers attention at the moment of selecting relevant or critical information (Engström et al., 2013). Activities are dynamically allocated and distributed between the available resources depending on its complexity and driver expectation of future scenario changes. The allocation of attention is defined as a gradually learned skill which varies from one person to other and depends on the exposure to different environmental driving scenarios.

Adaptive attention is also linked to the risk perception of each person. Drivers tend to adapt their driving activities allocation of resources inside a comfort zone. Engstrom uses an alternative term named safety margins defined as an spatio-temporal distance to a safety zone boundary.

Brown and Groeger explained that the risk perception is affected mainly by two inputs: information regarding the potential hazards in the traffic environment and ability of the driver to prevent those potential hazards from being transformed in accidents (Brown and Groeger, 1988). These two inputs can be seen as individual values associated to driving skills and driving experiences of each person.

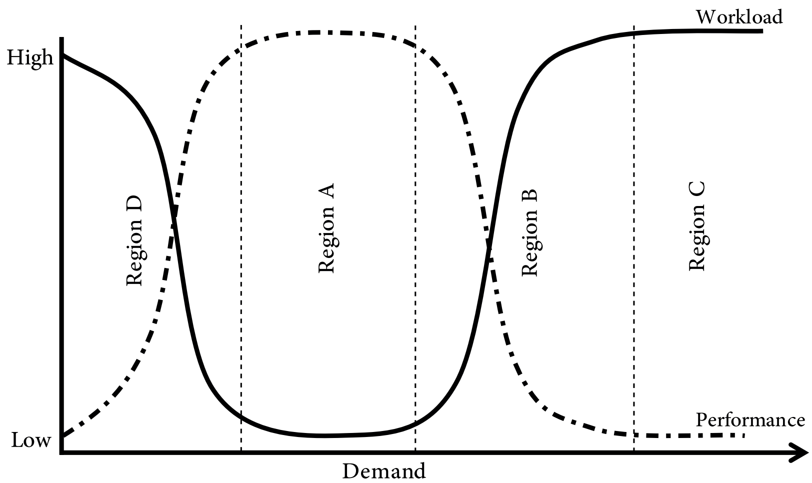
Inattention is a common focus of study in multiple-task scenarios. However, other researchers prefer an alternative and somewhat different concept for evaluating individual capabilities of performing properly a given task; i.e., mental workload or cognitive load. In 1982, O'Donnel (as cited in (Sofia and Pereira, 2009)) defined mental workload as a multidimensional interaction between: tasks and demands of the system, operator capabilities, experience and subjective performance criteria. Therefore, if a driver is mainly executing driving related tasks, it is expected that most of the demands be associated to driving task and therefore its workload be given as a function of the driving scenario.

There is not a general concept for workload, and it was almost unknown until 1970s when vehicles became integrating more complex system making available more services and new sources of information. As an overall concept, workload may be understood as a compromise between the mental demand required for executing a task and the capacity of that individual to meet these demands and produce and adequate level of performance.

Since the same level of demand not always produce the same reaction in different subjects, Schelgel in 1993 (Schlegel, 1993) introduced the concept of effort related to the capabilities, motivation, strategies, experience and also mood applied by a subject in order to complete successfully a task. Effort is as a voluntary process controlled by the operator, while mental workload is determined by the complexity of a task. In this regard, drivers can adapt their driving behaviour as a strategic solution to overcome multitasking. For example, experiments in simulators have revealed than drivers reduce speed when looking up telephone numbers (Brookhuis and De Waard, 2010).

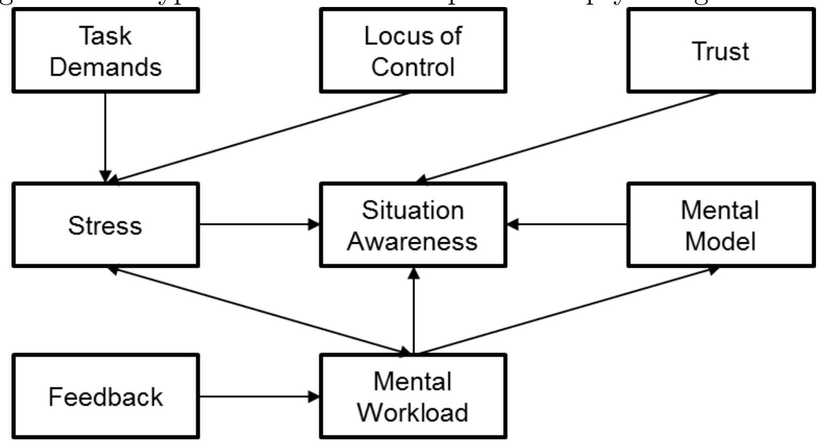
In 1996, Waard (De Waard, 1996) defines a relationship between the levels of performance/workload resulting of executing tasks associated to different levels of demand, see Figure 2.11. He differentiates 4 regions depending on the demand requirements of the tasks: *Region D*, where the performance is affected as a consequence of the monotony of a task; *Region A*, where it is achieved an optimal performance and *Region B* and *Region C* where the effect of increasing the task demand affects the performance. Waard's model is very useful for understanding that monotonous driving scenarios have also a negative effect over drivers performance and could also jeopardize safety.

Figure 2.11: Workload/Performance vs task demand (De Waard).



Other concepts are also useful for understanding the process in which the mind undertake the driving activity. (Stanton et al., 2015) presents a very intuitive framework that summarize how psychological factor are interconnected, see Figure (2.12. In this framework, *Feedback* received from the environment are the principal activator of the process and have a direct impact on the *Mental Workload*. *Situation Awareness* is presented as the central core of the framework and is affected by many factors as *Trust*, *Mental Model* (what individuals understand) and *Stress* levels. *Stress*, is influenced by the *Task Demands* and *Locus of Control* (referred to individuals' attribution of events to internal or external factors). In this framework, the level of *Situation Awareness* is presented as the main core in driving activity, therefore any change in its level is expected to affect the driving performance.

Figure 2.12: Hypothesised relationship between psychological factors.



Source: Human factors in automotive engineering and technology (Stanton et al., 2015).

2.2.4 Measuring drivers' workload

Different workload estimation techniques have been proposed and tested over time. Commonly used techniques are as follows: physiological evaluations, subjective measures, performance evaluation on primary and secondary tasks, and a consultation table definition.

As its name indicates, physiological evaluations get the information from involuntary body responses or variability due to different stress scenarios. Some examples are given on the basis of biometrical measures such as heart rate, brain activity, eye tracking, skin conductance and facial muscle tone. Several works have validated the existing relation between physiological measures and cognitive load. For example, Beatty demonstrated than when people are performing a complex task (cognitive load increment) their pupils dilate; this physiological reaction is called *task evoked pupillary response* (Beatty, 1982). It has also been demonstrated that the blink rate reduces when subjects perform high visual demand tasks and it increases when subjects perform high mental workload tasks without visual components (Recarte et al., 2008).

The drawbacks of cognitive load calculation based on physiological measures are given by the numerous factors that also affect the physiological response of subjects and are not related to cognitive load; Some examples are subject stress, health, user physical activity and environment variables (e.g, temperature and humidity) Miller (2001). In other study is stated that when using heart rate as a workload indicator, much interference is added due to emotional strains (Jahn et al., 2005).

Subjective measures, also known as self-report measures, involve the evaluation of the mental workload perceived subjectively by each individual. The main advantages of this technique are the low cost and easy implementation, while the drawbacks reside in the fact that users may have different perceptions of risk or be affected by personal stress rather than to driving-related tasks. De Waard exposed that no one is able to provide a better judgement with respect to experienced workload than the same person (De Waard, 1996).

Some of the most frequently used subjective measure techniques are: National Aeronautics and Space Administration-Task Load Index (NASA-TLX), which provide an overall workload score based on a weighted average of 6 sub-scales (Mental demand, physical demand, temporal demand, performance, effort and frustration level) (Hart and Staveland, 1988); Subjective Workload Assessment Technique (SWAT), which combines ratings of three different scales (time load, mental effort load and psychological stress load) (Hart and Staveland, 1988); Modified Cooper-Harperand (MCH) created for quantifying how an airplain pilot's workload affected task performance based upon controllability, workload, and attainable performance goals (Jr and Cooper, 1984), and the Rating Scale Mental Effort which is unidimensional and measures mental effort (RSME) (Harper and Cooper, 1986). Several works have used subjective measures for calculating workload, (Teh et al., 2014) use workload and driving performance for measuring the effect of traffic complexity.

Performance evaluations on primary and secondary tasks are also commonly used. Primary task performance metrics are related to driving in terms of lateral control, steering wheel pattern, longitudinal control, speed maintenance, brake pedal pattern and distance from other cars. The performance executing secondary tasks is measured in terms of accuracy and the time required to complete the secondary task. Another commonly used technique is called the "Peripheral Detection Task" (PDT), which consists in measuring the response time to randomly activated events (Harms and Patten, 2003). As regards response time, Dewar and Olson (Dewar and Olson, 2002) found that the majority of research indicates that it shows a range of variation between 0.75 and 1.5 seconds, when the driver is easily able to detect and identify hazards or problematic situations.

With respect to Wicken's Multiple-Resource Model, the PDT technique appears to provide a good indicator of driver workload depending on the driver scenario. The PDT passes through the same channel as the majority of common driving tasks causing interference in regular driving behaviour, therefore any difficulty or loss of attention when addressing the driving scenario will result in a slower response time to PDT events. A comparison of the benefits of employing physiological vs subjective measurement techniques was undertaken by Jahn given as conclusion that the PDT is sensitive to workload changes and provides a better indicator than physiological data (Jahn et al., 2005). Burns, in a different study found that the visual workload of drivers, as measured by a PDT, showed significant different across situations, road type and support levels (Burns et al., 2000).

The latter technique consists in creating a consultation table in which data concerning the user and the scenario are gathered for forecasting assumptions of driver behaviour based on previously known effects. This consists in a categorization of the driving complexity based on statistics from the drivers' age, gender, experience and driving style, together with that of the road based on the traffic situation, road geometry and weather conditions. Momentary workload is estimated mixing all these data and linking their effects.

Previous researches have implemented real-time monitoring systems for estimating driving complexity and its impact on the driver. (Amditis et al., 2010) describe one of these approaches in which the estimation is based on three parameters: Driver, Vehicle and Environment (DVE). The Driver parameter is associated with driver conditions such as personality, experience and attention state; the Environment parameter is related to the external driving scenario such as weather, traffic and road characteristics; and the Vehicle parameter provides data concerning to dynamic driving conditions (such as yaw rate, speed, acceleration, brakes), driving aids (Adaptive Cruise Control (ACC), Forward Collision Control (FCC)) and secondary In-Vehicle-Information-Systems (IVIS) interactions.

Several variables are grouped and associated to seven parameters, these parameters are then transformed into a three-dimensional vector, *Driver-Vehicle-Environment Vector*, which represents the given scenario complexity. The study justify the use of a vector instead of a single value since it would allow Human-Machine-Interface designers a more clear understanding of the scenario to define more uses cases depending each variable.

As a summary of this subsection, the more common metrics that can be used for assessing mental workload are presented in Table 2.5. Metrics are classified in four main groups (Physiological, Subjective, Performance, Environmental and Profile) and categorized as intrusive or not depending on how the measurement procedure affects the driving behaviour.

Table 2.5: Metrics that may be used for calculating mental workload in driving scenarios.

Type of Metric	Metric	Measurement technique	Intrusiveness
Physiological	Visual behaviour (Glances / winks patterns)	Wearable camera/ External camera	Yes/No
	Heart rate variance	Heart Rate Monitor	Yes/ No(Wearable)
	Brain activity	EEG	Yes
	Muscle tone	Muscle Tension Sensors	Yes
	Skin conductance	Psychogalvanic Reflex (PGR)	No
Subjective	Level of difficulty perceived	Questionnaire	Yes
	Level of distraction felt	Questionnaire	Yes
	Motivation	Questionnaire	Yes
Performance (Primary-Tasks)	Longitudinal control	Vehicle system	No
	Lateral control	Vehicle system	No
	Reaction time	Vehicle system/ Questionnaire	No/Yes
	Navigation errors	Navigation system	No
Performance (Secondary-Tasks)	Error while performing a secondary task	Questionnaire	No
	Time to complete tasks	Questionnaire	No
Environmental	Traffic level	Navigation system	No
	Road difficulty	Navigation system	No
	Environmental condition	Vehicle sensors/Online data	No
Profile	Profile information	Questionnaires	No

2.2.5 Summary

A list of factors, classified as endogenous and exogenous, that affect the driving performance and/or the driving behaviour were presented in this section. Due to randomness of crash accidents and the great amount of parameters that may increase the change of being involved in a crash, it becomes difficult to define the risk of accidents as a unique function resulting from all these factors.

Most of the research is based on statistical data or conclusions reached trough the performing of driving performance test and subjective evaluations. In case of tests, the driver behaviour could be conditioned by the supervision of the activity and the driving behaviour may differ to the followed under non-supervised situations. However, enough empirical data is available to make overall assumption about how driving conditions have an impact on road safety.

As a summary of the section, a table containing the main relevant factors that affect driver performance is included [5]. Table 2.6 shows how these different factors, previously exposed and classified as exogenous and endogenous, affect the risk of being involved in a crash accident. The *Conclusion* column shows the degree in which each parameter is expected to affect road safety from the higher to the lower according to the referenced studies. This table gives an idea about the large amount of factors that may affect driving performance and what must be considered for creating a model capable estimating an overall value of driving complexity.

A column indicating the data input type used in each research for concluding the risk associated to each factor is also included. This shows how most of the conclusions are reached using statistical data. These methods are common for assessing risks given the many variables that may increase the risk and uncertainty of crashes.

The measurement of drivers workload is vastly used for evaluating the impact of new technologies on road safety. As previously commented, a strong relation between the per-

[5]More detailed information about different crash statistics is presented in the Appendix Section (A.

formance executing a task and driver workload level exists. This implies that an accurate measurement of workload while driving could provide relevant clues about which stimulus have higher impact on driving performance.

Several technical has been proposed for measuring workload, some of them are highly intrusive which may affect the accuracy of the results. Regarding previous studies, the most recommendable techniques are based on subjective perception of participants (self-assessment) and performance in the execution of primary or secondary tasks.

Table 2.6: Summary of the influence of exogenous and endogenous factors on road safety.

Factor Type	Factor	Conclusion (high risk (up), less risk(down))	Data input type	Source
Exogenous	Road Area	non-motorway urban areas motorway	Statistics	(European Commission, 2015a)
	Road Area	urban areas rural roads motorway	SSR	(Östlund et al., 2004)
	Road Area	rural motorway urban	Irregularities in speed	(Östlund et al., 2004)
	Road Area	urban others	Hear Rate variability	(Östlund et al., 2004)
	Road Area	urban others	RT	(Östlund et al., 2004)
	Road Topology	roads with curves straight roads	SSR, DP and RT	(Östlund et al., 2004)
	Road Junctions	non-junction road crossroad T or Y junction rest of types	Statistics	(European Commission, 2015a)
	Number of lanes in road	more than three lanes two lane roads single lane flows	Statistics	(Highway and Safety, 2008)
	Manoeuvre previous to crash	going straight negotiating a curve stopped in a traffic light rest	Statistics	(Highway and Safety, 2008)
	Traffic level	high traffic density low traffic density	SSR and DP	(Teh et al., 2014)
	Presence of Heavy Good Vehicle	presence of HGV non presence of HVG	SSR and DP	(De Waard et al., 2008)
	Weather, (precipitations)	snow precipitations dry weather	Statistics	(Andrey et al., 2001)
	Weather, (sunrises and sunsets)	sunrise or sunset other part of the day	Other researches	(Institute for Road Safety Research, 2012)
	Weather, (temperature)	high temperatures low temperature (*no ice on road)	Medical researches	(Institute for Road Safety Research, 2012)
	Presence of pedestrian, cyclists...	presence no presence	SSR	-
Endogenous	Internal distractions	existing non existing	Statistics	(Highway and Safety, 2008)
	External distractions	existing non existing	Statistics	(Highway and Safety, 2008)
	Age	young drivers eldery drivers rest	Statistics	(Lancaster and Ward, 2002) (Co-operation, 2007)
	Gender	men women	Statistics	(Co-operation, 2007) (Lancaster and Ward, 2002)
	Driving experience	low driving eperience high driving experience	Statistics	(Lancaster and Ward, 2002)
	Personality	high values of “Sensation - Seeking” low values of “Sensation - Seeking”	Statistics	(Road, 1992), (Beirness, 1993)
	Physiology	drivers with physical impairement rest	Statistics	(Lancaster and Ward, 2002)
	Stress	drivers stressed rest	Statistics	(Lancaster and Ward, 2002)
	Fatigue	drivers fatigued rest	Statistics	(Hs, 2013)
	Speeding	speeding no speeding	Statistics	(European Commission and Energy, 2009) (Nilsson, 2004)
	Alcohol/drug influence	positive negative	Statistics	(Lancaster and Ward, 2002)
	Situation Awareness (SA)	low SA high SA	Studies	(Stanton et al., 2015)

*Abrv: SSR (Subjective Self-Reports), DP (Driver Performance); RT (Reaction Times).

2.3 Adaptive HMI, Previous Works

This section presents an outline of the most relevant works and research with regards to adaptive HMIs for infotainment systems. The majority of these researches are focused in defining schedulers for delaying information depending the driving scenario complexity, and in the adaptation of the interface applying minor changes on the design according to some predefined conditions in the scenario.

2.3.1 Generic Intelligent Driver Support System (GIDS)

A reference program devoted to the improvement of road safety was undertaken between 1989 and 1992 by the European Commission program. The first part of the project was called DRIVE I, and subsequently DRIVE II was developed between 1992 and 1994. These programs where aimed at facilitating the interaction of drivers with the infotainment system embedded in vehicles. One of the most recognized works developed during DRIVE I was the Generic Intelligent Driver Support System (GIDS) project (summarized in (Baber, 1994) and (Michon and Smiley, 1993)).

The overall objective of the GIDS project was to determine design standards for an intelligent co-driver system that maximized the requirements of information regarding the performance capabilities of drivers. The system was composed by several sensors connected to a workload estimator that informed about the current driver workload. The estimation was sent to a scheduler in charge of delaying or suppressing any presentation of information to the driver, and was also used by an adaptive interface manager which applied some basic restrictions on certain functionalities.

Whether or not a particular message should be presented or some functionalities should be restricted to the driver depended on the comparison between the observed behaviour and the required behaviour, and on the driver's needs and intentions. For this purpose, the modelling was based on a simple "look-up table" created according to Wickens's multiple resource theory.

A summary of these "look-up" table is presented in Table 2.7, here the presentation of information is classified according to the modality of information (e.g. Visual, Auditory, Cognitive, and Tactile). A priority level is assigned to each functionalities or presentations of information, where the maximum value is 6 (high priority) and the minimum is 1 (low priority). In case two tasks interfere in the same modality channels this value is used as reference for delaying, restricting or adapting the notification e.g, instead of presenting a long message when the workload it could be shortened.

The driving scenario was monitored at a sample time of 0.1 seconds and data were obtained from sensors such as a navigation system or a collision avoidance systems. The road scenario was separated in a series of segments where each one was classified according to the traffic level and a set of predefined manoeuvres such as: following the road, turning, negotiating an intersection, lane changing, overtaking, waiting a traffic light, among others.

In the project, the incorporation of a profile memorizing feature in which driven patterns of different drivers are stored was also proposed. Profile data could be portable (they propose the use of an smart card) in a way that the system can recognize each driver and propose driving alternatives and adapted guidance systems depending on the scenario. Moreover, the intelligent system was proposed to be flexible and that it could continuously learn from user behaviours and expertises, so the system can provide more information once the driver had more experience in a particular route.

Table 2.7: Driver interaction clusters as a function of input/output modality and priority.

Interaction Cluster	Highest priority allowed	System to Driver	Driver to System
1. Route Guidance			
a. entering destination	*	visual	
b. on-line guidance	3	auditory/visual	
c. asking route information	*		touch screen
d. route information presentation	*	visual	
2. Collision avoidance			
a. warning the driver	6	accelerator input	
3. Control Support			
a. install lane keeping support	*	visual	keyboard
b. lane keeping support	5	kinesthesic on the wheel	
c. speed and highway support	6	accelerator input	
4. Performance evaluator			
a. installing	*	visual	touch screen
b. scanning driver performance	*		
c. presenting feedback info	*	auditory/visual	touch screen
5. GIDS installer			
a. setting preferences	*	visual	keyboard
6. Repeat Last message	Depends	auditory/visual	switches/buttons
7. Telephone			
a. dialing	1	switches/buttons	switches/buttons
b. having conversation	2	auditory	speech
8. Stereo			
a. tuning	1	auditory	switches/buttons
b. volume control	2	auditory	switches/buttons
c. changing cassette	1	tactile/auditory	manual

Source: GIDS (Hoedemacker et al., 2002).

There were carried out acceptance test using a driving simulator. From these tests, results shown that 60-70% of the subjects considered the system useful, 80% considered the system would enhance safety, but only about a half of the total said they would buy a system like that. This project also suggested that traffic situation is the major determinant of the driver workload.

2.3.2 Application of Real-time Intelligent Aid for Driving and Navigation Enhancement (ARIADNE)

In the course of DRIVE II, the Application of Real-time Intelligent Aid for Driving and Navigation Enhancement (ARIADNE) was developed as a further step of GIDS (European Commission Project: DRIVE II, 1993). This project developed between 1992-1994, integrated the GIDS software into a Core System which consisted of three main modules: Manoeuvring and Control Support Module, Scheduler and Workload Estimator. The system was designed to control information sources and present these to the driver in a suitable manner. This improved system took into account more urban situations and road types to be used as inputs for taking decisions.

It was exposed that trying to measure the workload for all possible driving situations may be virtually impossible due to the large amount of parameters that characterize the driving task. As an alternative, a categorization of driving situations based on few specific characteristics was proposed. In general, the workload estimator consisted of a simple look-up table, where inputs from different sensors where evaluated and translated to recognized driving situations. Inputs used consisted mainly of the evaluation results of tactical tasks, which are more loading than control level tasks.

The workload indicator was used to manage which messages could be presented to the driver depending on their complexity. According to GIDS researches, traffic situation (e.g. traffic level, pedestrians or cyclists present) is considered the most relevant variable when assessing workload. Other important variable taken into account were the type of situation, driver experience and driver age.

Interesting conclusions were reached regarding the identification of the optimal way to present information through the use of messages. Navigation messages were found to have a lower impact on workload than other messages such as radio messages and a telephone call; exceptions were navigational messages in relation to a roundabout and approaching an intersection. It was also identified the optimum time interval between sequential presentations of information, it was predicted that the longer is the interval between messages, the lesser is the impact on drivers' ability to drive safely; as result the minimum recommended time between two auditory messages is three seconds (M. et al., 2002).

The research also evaluated drivers adaptation to the system after using it for a while and gaining experience. The hypothesis was that participants did not have time of adapting to the system and therefore driving performance or subjective feedbacks may not reflect the real impact of the system. A higher understanding of the system was expected to reduce the amount of unexpected behaviours and may show the real benefits of the proposal. It was traced the learning evolution of subjects with the support system over a total period of about 10 hours of driving on several types of road. Results provided little evidence for the existence of 'early indicators' of later adaptation to the support system.

2.3.3 Generic Evaluation Methodology for integrated driver support applications (GEM)

Project founded by the initiative DRIVE II between 1994 and 1995 and further evolution of the previous projects GIDS and ARIADNE. Its purpose was to develop a methodology for assessing how IVIS affect driving performance. The research was oriented to evaluate the impact of different applications on drivers workload, and how upcoming technologies such as Head-Up Displays could enhance the benefits with regards to driving performance (M. et al., 2002) and (Risk et al., 1994).

An extensive evaluation of different methods employed for assessing visual and cognitive workload was performed during the execution of the project. Its main purpose was to identify which methods provide a better indication of workload peaks in different scenarios. Table 2.8 shows a list of metrics compared in terms of their sensitivity to visual or cognitive workload. It is seen that the best indicators of visual load and mental load are the evaluation of secondary tasks, and subjective assessments as SWAT or RSME. Physiological indicators such as heart rate and eye blink do not seem to be sensitive to workload variation. As an exception, the skin conductance variability appears to be slightly correlated to mental workload.

Three applications categories were defined in terms of their priority in relation to driving tasks: Category 1 contains applications supporting primary driving task (lane support, collision avoidance system), Category 2 includes applications supporting important secondary functions (route guidance, traffic information, vehicle status), and finally the third category is related to less important secondary functions (e.g. radio, entertainment, trip information).

Table 2.8: Sensitivity of parameters to peaks of visual and mental loads.

Parameter	Visual load	Mental load
Steering wheel reversal rate [a]	Reasonable	Poor
Secondary task	High	High
SWAT	High	Reasonable
RSME	High	High
Skin Conductance	None	Reasonable
Hear rate variability	Poor	Poor
Hear rate (Interbeat interval)	None	None
Driving speed	None	None
Eye blinks	-	None

[a] Times per minute that the direction of steering wheel movement was reversed.
Source: TNO Factors (M. et al., 2002).

Infotainment interfaces were divide into two broad categories: (1) input devices as voice commands, touchscreen, softkeys, conventional controls, and (2) output devices as Head-Down Displays, Head-Up Displays, voice messages, tones, gas pedal (vibration), steering wheel (torque). Different devices were evaluated in terms of two parameters: the amount of time they were used and the frequency of use. Overall conclusions of this project do not offer clear estimations about optimal interfaces for presenting different information. Nevertheless, it did provide a remarkable background about suitable methods for measuring the workload while driving.

2.3.4 Communication Multimedia UNit Inside CAR (COMUNICAR)

The COmmunication Multimedia UNit Inside CAR (COMUNICAR) project was developed between 2000 and 2002. Its main goal was to create a system that managed the flow of incoming messages to the driver, this system regulated what, when and how the information should be presented regarding driver workload. The main purpose was oriented to provide information to the driver trough different sensory channels (Amditis et al., 2005).

The management of information presented to the driver was in charge of a module called Information Manager. The inputs used for taking decisions were based on data from four sources: the environment (visibility, humidity, ice detections), the traffic status around the car, a Driver Workload Estimator (DWE) developed by Metravib Inc., and input devices such as mouse or keyboards. From all these sources, an *Index of Risk* (IoR), represented by a priority level ranking from 1 (low risk) to 4 (high risk), was computed.

The DWE computed a three-level warning output based on 4 parameters coming from the clutch, brake, steering wheel, and blinkers. The Information Manager was in charge of controlling entertainment functions (Radio, CD,etc.), vehicle information (Speed, RPMs, etc.), telematic functions (navigation, messaging, phone, etc.) and Advanced Driver Assistant Systems (frontal collision warning, lane warning, speed adoption).

In a similar manner to GIDS and ARIADNE, the management of information was defined trough a set of predefined rules. Each message was assigned a priority level from 1 (high) to 4(low) and the IM was in charge of filtering those messages with a lower priority than the allowed by the actual *Index of Risk* (IoR).

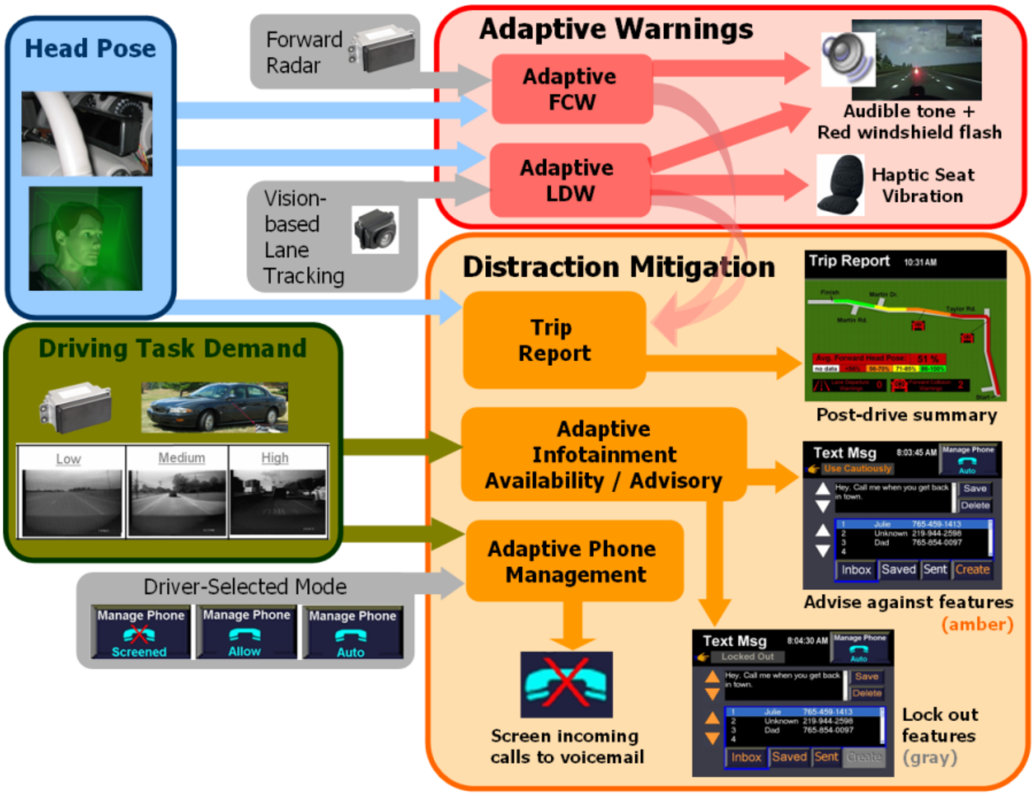
2.3.5 SAfety VEhicles using adaptive Interface Technologies (SAVE-IT)

One of the more complete programs devoted to create adaptive infotainment systems was the SAfety VEhicles using adaptive Interface Technologies program (SAVE-IT) which started in 2002. It was a 5-year research program sponsored by the NHTSA and administered by The National Transportation Systems Center (NTS) and Volpe (Brown et al., 2007). The program aimed to reduce distraction-related crashes by implementing adaptive countermeasures and monitoring the driver state.

The adaptive system proposed in SAVE-IT was mainly focused in the integration of visual, auditory and haptic warnings in response to expected threats or hazards. A basic restriction mechanism that locked out certain features such as incoming calls by disabling some buttons available in the interface was also part of the system. The evaluation and validation of the system was executed trough questionnaires to participants after they were requested to test the system driving in a simulator for two hours. Overall results were positive and participants reported good feedbacks with regards to user acceptance.

Figure 2.13 shows a complete scheme of the proposed system developed in the SAVE-IT project. The inputs to the system for estimating the driving complexity are computed from the driver behaviour (Head Pose) and some driving performance metrics (Traffic detection using a radar and lane deviation). These variables are then used as reference to generate adaptive warnings to the driver as well to offer distraction mitigations. Changes in the HMI system are based on warnings, restrictions of available functions and re-scheduling of incoming calls.

Figure 2.13: SAVE-IT Scheme.



Source: SAVE-IT project (Brown et al., 2007).

2.3.6 Adaptive Integrated Driver-vehicle interfacE (AIDE)

Another significant research in the area of adaptive interfaces was the Adaptive Integrated Driver-vehicle interfacE (AIDE) which started in 2004. This European program involved 31 partners and consisted of several research phases aimed at estimating driver workload and defining guidelines for regulating the presentation of information depending on Driver-Vehicle-Environment (DVE) variables. (Amditis et al., 2005).

To address this objective, the architecture proposed consisted on different modules oriented to monitor the driver state, analyse the driving environment, and adapt the HMI. The architecture had a centralized structure and all decisions were taken by a central module called Adaptive Interface Module (AIM); this module had a full view of the current scenario and its main role was to avoid the interference between different pieces of information. Moreover, nomadic devices like personal portable phones or PDA's were also taken into account in the architecture; notifications or incoming calls from external devices could be inputs to the AIM representing increments in the driver state workload. The modules connected to the AIM are the following:

- Driver Characteristic module (DC): Manage profile information about the different drivers. The system is preconfigured with certain profile parameters and a *Customization Agent* is in charge of monitoring and learning about the user behaviour and preferences. These data can later be used to predict user needs and provide more adequately the functionalities.
- Availability of the Driver Estimation module (ADES): Monitor primary driving tasks which reflect the driver basic interaction with vehicle controls as pedals, steering wheel, blinkers, etc.
- Cockpit activity assessment module (CAA): Monitor information about driver state in regards of his attention level. Visual demand patterns are monitored trough the integration of eye and head tracking devices.
- Driver state degradation module (DSD): Supervise the fatigue and hypo-vigilance of drivers.
- Traffic and environmental assessment (TERA): Oriented to calculate the level of risk due to traffic and environmental conditions.

2.3.7 Summary

This sections exposes a summary of the most notable works devoted to the design and development of infotainment systems. The main premise of these projects was the definition of schedulers that controlled the timely presentation of information depending on the scenario complexity or driver workload. Neither of them directly addressed an adaptive interface design, which is the goal of the current dissertation. This background serves as baseline for the development of the proposed concept; many of the methods and techniques implemented are based on these approaches.

2.4 Machine Learning Methods

Engineering and scientific researches usually need to interpret large amounts of data, sometimes without clues about the expected outcome. For these cases, appropriated tools that ease and accelerate the analysis are required. The purpose of data analysis can be summarized in: the extraction of valuable information about the nature of the data and the creation of models capable of fitting its behaviour and making predictions on new inputs (Breiman, 2001b). For this purpose, data mining methods provide a very good toolbox for dealing with large amount of data. This section exposes some of the tools and methods applied during the development of the project concerning data analysis.

The process consisting in extracting information of a database and finding hidden patterns is kwnon as Knowledge Discovery in Database (KDD). KDD is an iterative process consisting of several stages that can be summarized in: understanding of the problem, processing of data collected, and evaluation of results. Data mining is the core stage of this KDD process and it deals with the processing and analysis of the data.

A far important stage in the KDD process is the analysis of data after the processing stage. This phase is oriented to understand the results and identify the benefits in relation to the study, the final outcome could allow to create overall rules that characterize the data and can be generalized for the complete dataset. Results can be quantified trough objective measures resulting from statistical reports, or subjective analysis that offer a more personal perception about the outcome and its meaning. Both are complementary and used wisely depending on the type of data analysed.

Subjective appreciation plays an important role in KDD since the final purpose of the data mining process is to identify relevant rules and patterns related to the followed purpose or that bring interesting information about the dataset (Al-hegami, 2004). Therefore, subjective evaluations are highly associated to the *interestingness* of the rules depending personal appreciations. This *interestingness* is given as a function of three parameters: unexpectedness, actionability and novelty. Unexpectedness of a rule can be regarded when certain rule contradicts user belief about the domain, actionability of a rule exists when the rule found is useful for the purpose of the application and finally, novelty when the rule contributes to new knowledge.

According to Al-hegami, a KDD is considered successful if it provides the user previously unknown, useful and interesting knowledge (Al-hegami, 2004). The last two terms of this function are subjective measures and depend on the purpose of the study. Novelty plays an important role during the complete process of KDD, at the first stages it can help filtering unnecessary or irrelevant information helping reduce the dimensionality space, during the learning process rules can also be classified depending on the metrics that are expected to bring novel information, and finally in the analysis stage rules can be classified depending on the needs.

The concepts of data mining and machine learning are very related terms and sometimes are presented without distinction. Large discussions exists in this regards, and for the purpose of this study machines learning methods will be understood as a subset of data mining. Data mining can be seen as the complete toolbox that provide the mechanism to describe and characterize the nature of the data (descriptive data mining), while machine learning methods allows to create predictive models that fit the data and allow to estimate and predict outputs for fresh inputs (predictive data mining). In both cases, mathematical concepts are used for creating efficient algorithms that can deal with great amount of data.

A machine learning method basically needs a training dataset that is used as baseline for constructing the model. In this dataset, the rows that compose it are defined as training examples (it can also be understood as a previously known case). Each training example has a set of attributes that characterize it (input vector) and usually belong to a certain class (output).

Depending on the nature of the data, learning algorithms can be classified into supervised and unsupervised methods. Supervised learning is applied when the training dataset also contains information about the class they belong, therefore the learning method consist in adapting the model observing the deviation from the expected output. Unsupervised learning is used when the outputs of the training dataset are unknown. As a consequence, the algorithm must be able to identify the cluster that allows a categorization of the training set based on their features. Most of the machine learning methods developed are capable of dealing with these two modes.

Many machine learning methods are available and unfortunately there is no optimal model that offers the best result for all the possible datasets. Depending on the data, some may offer better performance than others and the opposite may occur for a different type of data. Moreover, depending on the metrics defined for measuring the performance the most suitable machine learning method may change for the same dataset.

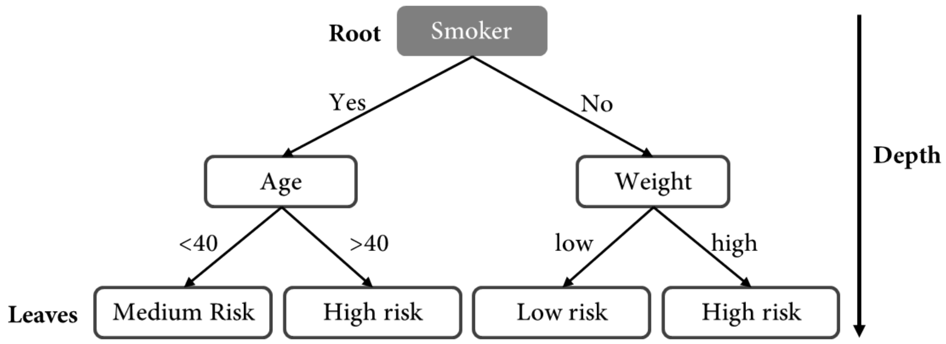
Some of the most relevant machine learning methods for the purpose of this research are summarized in the following subsections.

2.4.1 Decision Tree (DT) and Decision Tree Ensemble

Decision Trees are one of the most popular and initially created method for the classification of data in terms of different attributes. It is also one of the most understandable classification models since it is composed by a set of rules that can be graphically represented in a tree structure. This representation allows to easily recognize existing patterns and determine the relevance of features.

The structure of the tree is composed by nodes and arcs; nodes are labelled by an attribute name and arcs define the links between nodes depending certain threshold values. Figure 2.14 shows a graphical representation of a simple Decision Tree, the top node is called the root of the tree and bottom labels are called leaves (class feature). When predicting data, a Decision Tree is traversed in a top-down manner, following the arcs according to the input vector until finally getting to a leaf node which contains the classification expected for that case. Once the tree is built, parameters located in top layers have a higher influence in the characterization of the dataset and prediction of fresh data.

Figure 2.14: Decision Tree structure (example, non real, of risk of heart attack evaluation in term of two parameters).



A Decision Tree is constructed using algorithms based on statistical information of the dataset. These algorithms are capable of grouping elements in the dataset according to classes and attribute thresholds. The purpose followed is not only to correctly classify existing training examples in the dataset, but also to categorize unseen inputs. In order to achieve this, the Decision Tree must identify the relations between parameters. Moreover, data usually contains certain level of noise which may be not uniform distributed for all classes; in this regards, the algorithm must recognize misleading training examples.

As a consequence of the recursive partitioning of data while constructing the tree, the number of examples contained in each node decreases in deeper layers. It must be taken into account that complex models that categorize the training data very well may not be adequate for predicting unseen data, since the model is memorizing and not learning the overall patterns; this phenomena is known as overfitting. In order to solve this issue, many Decision Tree algorithms employ a post-tree creation mechanism in which branches and nodes located near the leaves are pruned. This new leaves resulting after the pruning are no longer pure nodes and do not contains only training examples of a same class but a mixture where the most popular class will be the result in predictive applications.

Several algorithms have been designed with the purpose of constructing Decision Trees, ID3 was one of the first proposals exposed by Quinlan in 1986 (QUINLAN, 1986). He later presented the algorithm C4.5 (Quinlan, 2014), which became one of the most well-known and commonly used algorithm. Some variants of this algorithm are SEE5, J48 and a further improvement named C5.0. The C4.5 algorithm assigns parameters to each node based on a purity measure (the purity is an indicator of the percentage of correct classifications of kwnon data after making the split of data). In this process, the purity measure is computed for the current node and all successors and this result is compared with previous parameters used. In C4.5, the information-theoretic entropy is used as purity measure (Information Gain). Other impurity-based criterion commonly used are Gini-Index, DKM Criterion and Gain Ratio.

In regards of processing time and performance, Decision Trees are very fast and several mechanism for paralleling the construction have been proposed. A fast and scalable decision-tree-based algorithm (SPRINT) that does not require that all or part of the dataset remain in memory has also been developed (Shafer et al., 1996). The algorithm has demonstrated to work well in parallel processing units which accelerate the time needed for creating the model.

An improvement in the accuracy and stability of the prediction can be achieved trough the construction of several Decision Trees from which the most common value resulting among all the trees is considered for the final result. This approach, that make use of several machine learning methods in order to increase the predictive performance is known as ensemble of methods. An ensemble of trees consists in using several Decision Trees which are training using different part of the dataset or different parameters. Some of the most common ensemble-tree based techniques are Bootstrap aggregation (Breiman, 1996), Boosting (Freund and Schapire, 1996) and Random Forest (Tin Kam Ho, 1995) and (Breiman, 2001a).

Bootstrap aggregation, also called bagging, consists in taking the base algorithm and invoking it many times with different training sets. As result, a group of diverse Decision Trees is obtained. Each training set is constructed sampling uniformly and with replacement (an element may appear multiple times in one sample) the complete dataset (S). Boosting is a method used for improving the performance of any learning algorithm; a very known boost algorithm is AdaBoost (Adaptive Boosting). The AdaBoost algorithm assigns a set of weights over the original dataset which are adaptively adjusted depending the accuracy of the learner, called "weak learner".

Random Forests create multiple trees randomly selecting a subspace of the feature space. For a given subspace of m dimensions, there are 2^m subspaces in which a Decision Tree can be constructed. All trees are training using the complete dataset. Nevertheless, the generalization will be different since not the same features are included in each tree. Random Forest are comparable to AdaBoost in terms of error rate, but it shows a great advantage when it respect to noise insensibility (Breiman, 2001a). Random Forest can also improve the performance not only selecting random features but also implementing other mechanisms as AdaBoost which take different training sets depending on their weight.

A comparison between different ensemble-tree based methods is presented in (Banfield et al., 2007) and (Dietterich, 2000). Examining different datasets, they found that boosting and Random Forest are statically better than bagging. Both, AdaBoost and Random Forest have similar prediction accuracy performance and are capable of overcoming the effect of noise. An advantage of Random Forest in comparison to AdaBoost is with respect to the tuning needed previous each training process, for AdaBoost this configuration is more complex than with Random Forest where the only parameters needed are the number of trees and their maximum depth.

2.4.2 Neural Networks (NN)

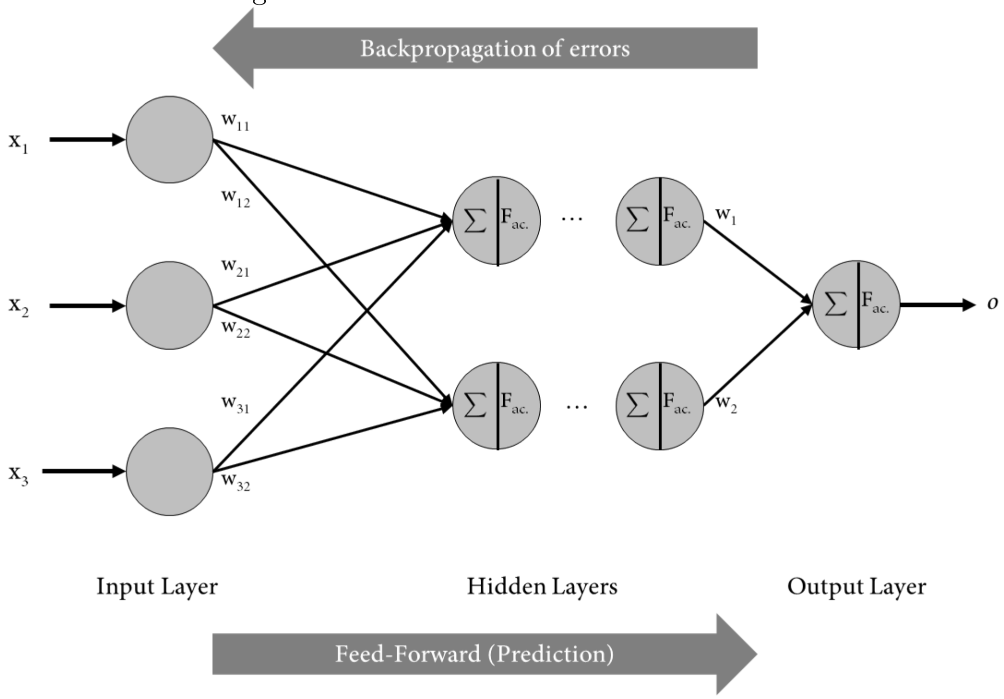
Neural Networks are a type of machine learning methods inspired in the nature of the brain, specially in the transmission mechanism of information between neurons. The first proposal of Artificial Neural Networks dates back to 1943 when Warren McCulloch and Walter Pitts presented the first model of artificial neurons (Riesel, 2007). From this point on, new and more sophisticated proposals in this regard have been presented.

Biological neurons are linked one to each other in a weighted way, these connections are called *synapses*. When a stimulus is detected by a sensory receptor, it is transmitted to the central nervous system and the signal is received by a neuron through the *dendrites*. This signal is electrically transmitted to following neurons via the *axon*, during this transmission stage the signal is altered by a biochemical process. The subsequent neuron receives several post-processed signals which are summarized or accumulated in a single pulse, depending on the value resulting (electrical potential) and certain threshold the neurons fire another pulse or not, thus the output is non-linearly dependent on previous signals (Riesel, 2007). But what is more interesting about the physiology of the brain is the way in which these interconnections between neurons are dynamically adjusted during the learning process, is this remarkable issue the true inspiration of this method. Artificial Neural Networks make use of mathematical concepts for simulating, in part, the transmission of information between neurons.

The feed-forward neural network was the first and simplest type of Artificial Neural Network devised. Here, the biological network is modelled representing the incoming stimulus by an input vector ($\vec{x}$), in the transmission stage the signal is weighted and transmitted to subsequent neurons, called nodes. Each node sum the weighted signals and if the result is above certain threshold (activation function) a fixed value is transmitted to further nodes (1 is usually used for indicating activation and -1 in case of deactivation). A feed-forward network is characterized for having a topology that has no closed paths (Rojas, 1996). Recurrent networks, on their counterpart, allows the creation of directed cycle connections between nodes.

Figure 2.15 shows a classical feed-forward network called Multi-Layer Perceptron (MLP). In a MLP three main layers are defined: (1) the input layer in which each input element of the incoming vector is received by a node, (2) the hidden layer where a predefined num-

Figure 2.15: Feed-Forward Neural Network.



ber of nodes receives the weighted inputs from the input layer and compare the resulting sum with a threshold, and (3) the output layer in which is returned the category of the prediction. The number of hidden layers and nodes contained in each one can be increased depending on the requirements of the application; a neural network composed by a large amount of hidden layers to the network is known as “deep neural network”.

The learning process of a neural network consists in the adaptation of links’ weights in order to reduce the error. The backward propagation of errors algorithm, abbreviated as Backpropagation, consists in an iterative process where the error function of the network is calculated in each step with respect to the weights of the network (randomly set at start). Then these weights are modified in order to achieve a reduction of the error. The squared error function of the network E is given by equation 2.2, where o_i is the predicted output for a training example, t_i is the expected target and p the total amount of training examples in the training set.

$$E = \frac{1}{2} \sum_{i=1}^{p} \parallel o_i - t_i \parallel^2 \tag{2.2}$$

Since each node contributes with a different weight, an error function which has as parameter the weight is needed. The output of single node o_j can be expressed as a result of the activation function $\varphi()$, given the weighted inputs coming from previous nodes $w_{kj}o_k$, as expressed in Equation 2.3. Several activation functions are used in MLPs, some of the most commons are Linear, Gaussian, Elliott and Sigmoid. From the collection of all the nodes’ contributions, o_j, a single value which represents the output of the network is obtained o_i.

$$o_j = \varphi\left(\sum_{k=1}^{n} w_{kj} o_k\right) \tag{2.3}$$

Because the error function E is calculated in terms of the weights in each node, it is continuous and differentiable. The next steps consists in minimizing its value by using an iterative process of gradient descent ∇E which allows to find the direction to scale each weight and reach a local minimum error. Each weight of the total l is updated by scaling with a predefined value, γ, called learning rate. Each weight is continuously updated using the increment presented in equation 2.4.

$$\nabla w_i = -\gamma \frac{\partial E}{\partial w_i} \quad \text{for i=1}, ...l \tag{2.4}$$

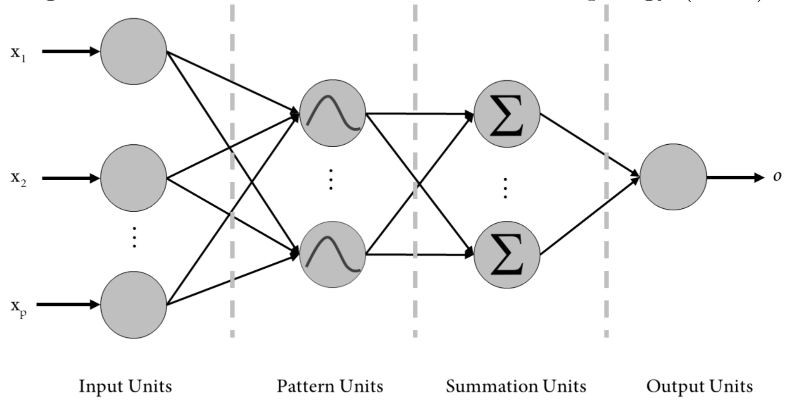
Backpropagation is a rather slow algorithm and many alternatives have been proposed in order to accelerate the learning process (Rojas, 1996). A notable proposal, called resilient backpropagation (Rprop), was presented by Martin Riedmiller and Heinrich Braun in 1992 (Riedmiller and Braun, 1992). This algorithm updates the weights just using the learning rate and the sign of the partial derivative of the error function with respect to each weight. Variants in the implementation of this algorithm were proposed in (Igel and Hüsken, 2000) and (Igel and Hüsken, 2003). Among all the variants (Rprop$^+$, Rprop$^-$, iRprop$^+$ and iRprop$^-$) the best performing algorithm in terms of training speed was demonstrated to be iRprop$^+$ (Igel and Hüsken, 2003).

Probabilistic Neural Networks (PNN)

Probabilistic Neural Networks can be seen as an evolution of Neural Networks and were introduced by Donald F. Specht in 1990 (Specht, 1990). He proposed a fixed neural network topology composed by four layers: an input layer, a pattern layer, a summation layer and an output layer. This fixed structure simplifies the model and reduces the uncertainty at the moment of choosing the more suitable topology. The technique proposed offered a great speed advantage in the learning phase; it was shown that for one application, the PNN was 200.000 times faster than back-propagation.

Figure 2.16 shows the organization of a PNN; Inputs Units are linked to independent variables composing the incoming vector, each Pattern Unit receives the input values from all the possible incomes, forms a dot product with a weight vector vector $Z = X \cdot W$ and performs a non-linear operation using an activation function resulting in a vector of probabilities for each class. Summation Units sum the results obtained from the pattern inputs that correspond to the category selected and finally the outputs units produce a binary output corresponding to one of the available classes. More than one output unit can be added for resolving multi-class problems with more than two categorical outputs.

Figure 2.16: Probabilistic Neural Network topology (PNN).



The training consists in finding the best vector of weights W that optimizes the classification accuracy. For this purpose several activation functions can be used by the Pattern Units. Specht proposed to use an exponential operator given by $exp[(Z_i - 1)/\sigma^2]$ instead of the Sigmoid function commonly used in neural networks. The smoothing parameter σ allows to adjust the fitness of the prediction; it was found that in practical problems small changes in the value did not affect the misclassification rate but did have a clear impact on training performance (Specht, 1990).

Overall, PNN can be seen as an improvements of Neural Networks in regards of training speed, accuracy and insensitivity to outliers. A disadvantage in comparison to Neural Networks is the higher requirement of memory needed for storing the trained model. These training algorithms do not detect redundant neurons and tend to expand the size of the hidden layer more than necessary.

(Berthold and Diamond, 1998) introduced a new method for constructing Probabilistic Neural Network based on a type or Radial Basis Function Network (RBFN) called Dynamic Decay Adjustment (DDA). The proposed algorithm introduces hidden nodes only when needed reducing the final size of the network. Moreover, the algorithm provides a method for avoiding over-training since it indicates when the training is complete. DDA distinguish between matching and conflicting neighbours in an area of conflict using two parameters, θ^- is used to avoid misclassification (lower values have a higher probability of being incorrectly classified) and θ^+ indicates the minimum probability of correct classification.

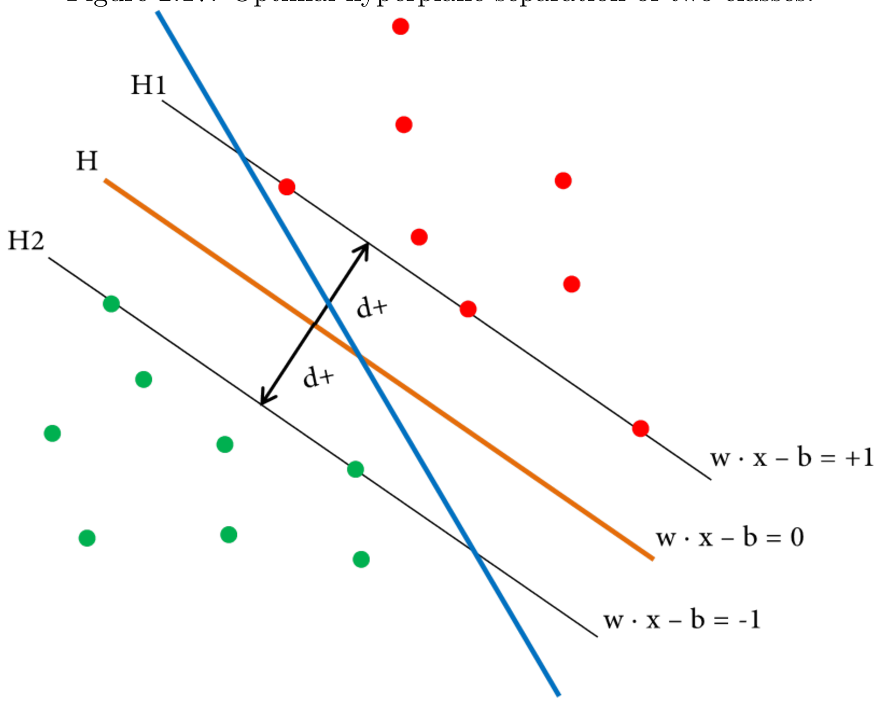
In summary, PNNs implementing DDA algorithm are characterized by a dynamic addition of neurons whenever necessary, a fast training (less than 5 epoch are needed to complete the training) and a low configuration needed since only two parameters are needed to be specified, which have been also demonstrated to be not critical. Also, the network size does not scale linearly dependent on the dataset size and allows dealing with redundant data. More important, as the probabilities of classifications are given as part of the result, practical applications can benefit taking additional decisions in regards of high or less accurate results.

2.4.3 Support Vector Machines (SVM)

Support Vector Machines, introduced in 1995 by Vladimir Vapnik (Cortes and Vapnik, 1995), rely in the mathematical concept of kernel functions. First machine learning methods classified elements belonging to linearly separable domains using simple hyperplanes, this was a large limitation to most of the problems since not all datasets can be linearly classified. As an improvement, a transformation in the dimensional space of the dataset was proposed trough the implementation of kernel functions (Crammer and Singer, 2001).

Kernels allow to map data to a higher dimensional space where data can be separable, this transformation can be done by using either linear transformations or more complex functions as polynomials, radial or sigmoid to cite some examples. Once data has been transformed, the problem is reduced to an optimization problem where we see to find the hyperplane that maximizes the geometric distance to the closest datapoints. Figure 2.17 shows two hyperplanes (blue and orange lines) that separate correctly two classes represented by red and green points. Nevertheless, just one of them (orange line) optimize the separation of the closets points of each class.

Figure 2.17: Optimal hyperplane separation of two classes.



The distance between the hyperplane (H) and one of the margins, H1, is given by equation 2.5, changing b by a negative the distance to H2 is obtained. In order to find the maximum distance between these two margins (given by $2/ \parallel w \parallel$), the value of $\parallel w \parallel$ must be minimized and as constrains none datapoint must be present between H1 and H2 ($y_i(w \cdot x_i + b) \geq 1$ for all datapoints). The principal method used for achieving this consists in employing Lagrange multipliers (α_i), from which is obtained that the vector w is a linear combination of the training examples (each example has d inputs and a class label of two values $y_i \in \{-1, 1\}$) and only the vectors associated to the closest datapoints, “Support Vectors”, contribute to its value.

$$\frac{\mid w \cdot x + b \mid}{\parallel w \parallel} = \frac{1}{\parallel w \parallel} \tag{2.5}$$

Supposing a more random distribution of points in the space, a complete separation of points may be impossible for the current domain space. A kernel transformation ($z = \phi(x)$)

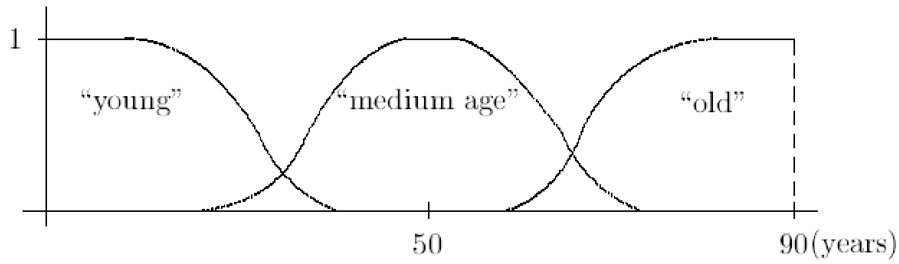
is then applied before starting the optimization problem for finding the best hyperplane. An issue in this regards is the identifications of the adequate transformation $\phi()$ since a complete transformation of the dataset and its further evaluation will probably lead to a large computing time. An overestimation of the dimension size will probably offer at least an hyperplane that separate all classes but it may fall in overfitting. And on the contrary, a transformation to a small dimension may not been enough for separating datapoints. SVMs overcome this issue thanks to a kernel trick; any time $\phi(x_a)$ appears, it does in a dot product with some other $\phi(x_b)$ and it is only needed to know the result $\phi(x_a) \cdot \phi(x_b)$. Kernels offer the advantage of directly getting this information.

Some of the most common kernels used are: lineal, gaussian, sigmoid, polynomial and radial. All of them are based on the same principles Nevertheless, depending on the distribution of points in the dataset some are faster creating the model and deal better with overfitting. A further improvement of SVM achieved the implementation of an algorithmic for a multi-class kernel based vector machine (Crammer and Singer, 2001). In comparison to previous approaches which reduced the multi-class problem in a binary one, the approach presented uses the definition of margins for multi-class problems and the optimization of a quadratic objective function.

2.4.4 Fuzzy rule leaner (Fuzzy Logic)

This method explore the concept of partial truth were results are expressed not only as boolean values but as a continuous range of real numbers between 0 and 1. This approach allows to find a more subjective analysis since values can be given in distance to a reference value. For example, the age of a person may be given in a range of values going from very young to very old instead of exact numeric values as can be seen in Figure 2.18.

Figure 2.18: Possible form of Fuzzy Logic result.



Source: (Novák et al., 2001).

Fuzzy Rule methods are very useful when the uncertainty of a response provides relevant information in the analysis and due to its graded approach may provide solution of some, classical non-solvable problems. Fuzzy Logic methods make use of linguistic variables to expose results such as very young, medium age, quite old, not old but young.

A very complete presentation with regards to Fuzzy Logics is exposed by Novak in (Novák et al., 2001). Two complementary facets of the indeterminacy in fuzzy logics are; uncertainty which is given by the lack of knowledge about the occurrence of an event and vaguensess expressed as the opposite of exactness and given by the nature of language for classifying elements in groups.

The mathematics behind fuzzy sets seeks an approximation for classifying objects in vague groups where natural language plays a very important role.

2.4.5 Summary

A large amount of machine learning algorithms have been created. Some of them are inspired in biological concepts as genetic algorithms, other are inspired in mathematical and geometrical concepts as logistic regressions, and clustering, and others in uncertain perceptions as Fuzzy Logic. A complete explanation of existing algorithms would be very extensive and more than one book will be needed to present all the information. This section was devoted to present a summary of the most common and recommended methods.

Apart from the certain mathematical concepts, all the algorithms have in common the creation of an adaptive model capable of estimating outputs. Models are created using adaptive training methods that take as reference kwnon information called training set. Here lies the importance of having enough and reliable amount of data.

A universal comparison between different machine learning algorithms will result in a conditional conclusion depending on the metrics employed and the expected goal. Each machine learning method has advantages and drawbacks depending on the dataset used and the practical application in which the model will be implemented (e.g. for image recognition, weather forecast, identifying heart failure, assurance policy classification). Therefore, a common practice to select the more suitable, consists in the evaluation of several algorithms for the available dataset, and select the one that adjust better to the requirements.

Some of the metrics that may be used for evaluating the performance of machine learning methods are: true positive rate (Sensitivity), false positive rate (Specificity), F1 Score, mean absolute error, mean squarred error, confusion matrix, size of the final model, training time and computational time for making predictions.

Chapter 3

Methodology

The project was developed in two main phases consisting of (1) the construction of a model capable of estimating driving complexity and (2) the definition and validation of an adaptive interface for in-vehicle-use based on the driving complexity estimation. This section exposes an introduction to the methodology undertaken in each phase to create the proposed system. It is also presented the set of tools, software, components and hardware that were used during the study.

The most adequate manner to start explaining the concept followed seems to be the presentation of graphical scheme of the proposed system. Figure 3.1 shows the overall concept where several variables that may serve as indicators of driving complexity are given as input to a model capable of calculating driving complexity. The output of the model is then used for adapting the presentation of information and interaction with the functionalities available trough the different interfaces connected to the vehicle. Input variables are profile data, physiological values, driving performance information, HMI interaction and environmental conditions.

Figure 3.1: Scheme representing goal of the proposed system.

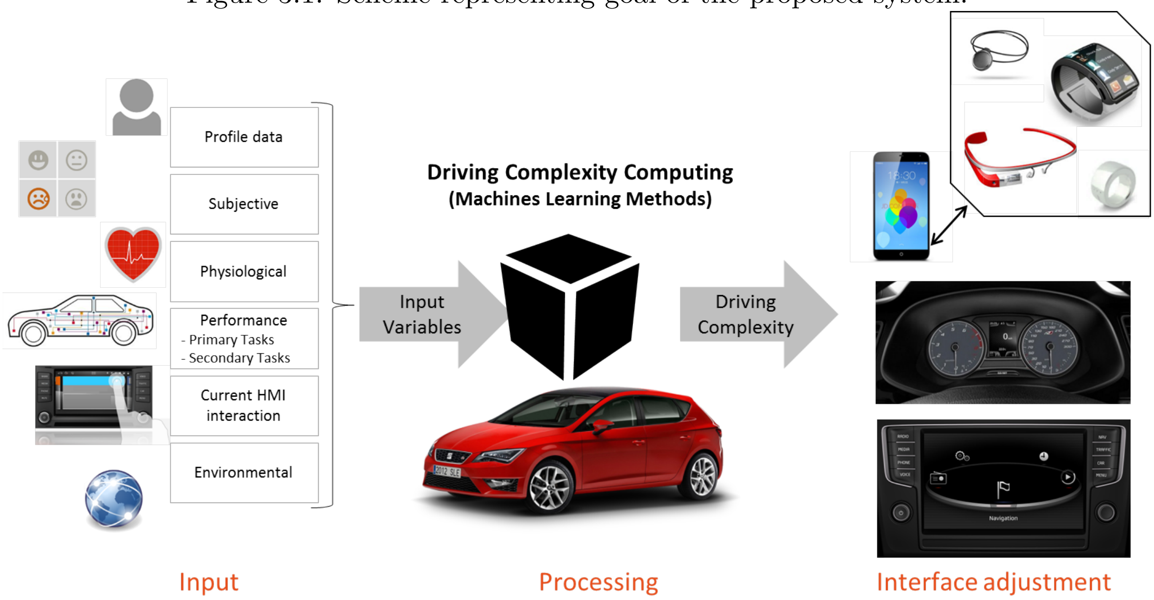
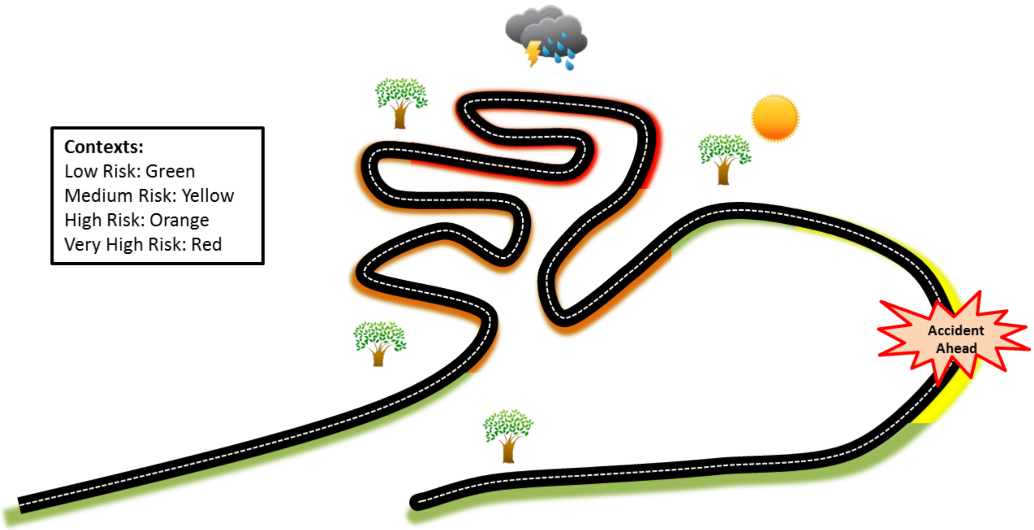


Figure 3.2 shows an example of how the driving complexity, represented by 4 values, can change depending on certain conditions, in this example only road type, traffic and weather conditions are considered. The first phase (green) could be seen as a highway without traffic where the user is allowed to interact with all the functionalities since the driving complexity is low. The second phase (yellow) contains several curves and therefore is expected an increment in the driving complexity causing that some Graphic User Interfaces (GUI) adaptations be applied. In the third phase (red), the weather conditions change and a due to a raining condition most of the functionalities are restricted or adapted to reduce driver workload to the maximum level possible. Finally, a yellow area is given a consequence of an accident ahead. In this case the user can be alerted.

Figure 3.2: Driving complexities computation example based on road and weather parameters.



The proposed system, in comparison to previous proposals, computes an estimation of the driving complexity based in larger amount of variables. This set of variables is expected to describe the driving complexity more accurately in more diverse conditions. Input information to the estimator can be obtained not only from the vehicle but also from several devices such as wearable, smartphones and even other vehicles or city sensors.

The black box responsible for making estimations is constructed using machine learning methods and it is based on data collected during different driving scenarios. This data are correlated to two feedbacks: users' subjective evaluation of the driving complexity and response time to events randomly activated while driving.

3.1 Architecture

The system is basically composed of sensors, user interfaces and processing components. In order to have a smooth and reliable communication between all components. Three architectures were considered: a centralized scheme were all the information passes through a central device, a distributed scheme were every device is self-managed and the processing load is shared among the connected devices, and finally a mixed scheme where the advantages of centralized and distributed solutions are exploited.

Centralized systems are useful when a central control for monitoring all the resources is needed. The main advantage is that one device with enough computing power can process all the information and other devices can be assigned less demanding operations (sensors and user interfaces without a complex logic). Moreover, in this scheme external devices are easy to update trough central deploys. The drawbacks of this scheme are the dependence of a central device which makes it not fault tolerant and scalability issues may come up when adding new functionalities in expansions.

A distributed approach copes with some of the drawbacks of central systems; it is fault tolerant since redundant devices can be used for performing the same tasks, a better management of the load can be achieved, and these systems can be scaled effortlessly. The problem of this approach comes out in case of using constrained devices in terms of energy or computation power since the overall performance of the network may be affected.

The proposed solution consists in implementing a mixed approach between centralized and distributed schemes. This architecture can guarantee a proper level of processing load taking into account constrained devices and can offer fault tolerant mechanism. In this scheme, a central device is proposed to execute most of the computing operations and the rest of connected devices may request or post information only to those devices that have or need some information.

Since the vehicle will always be a part of the architecture, this element is proposed to constitute the central core in the mixed approach. The vehicle system has wired access to most of the variable related to driving parameters and it can also offer information associated to weather conditions and driver interaction with infotainment systems. This reduce considerably the transmission of information with external devices.

The main core of the driving complexity algorithm is proposed to be installed in the vehicle; updates of the driving complexity value can be published by the vehicle towards the rest of connected devices. In this mixed approach, external devices are responsible of taking decisions in regards of how adapting their own interfaces depending the driving complexity. These devices can also perform some pre-processing operations before sending input information to the central unit in order to reduce its computational load.

In order to guarantee scalability and future integrations, the driving complexity estimation model must be light and should be upgradable in case new inputs are added. Devices should also be able to communicate between each other without the need of the central device, this is useful in case a device wants to make an interface change depending on certain functionality used in other device.

In this architecture, bandwidth overloads are not expected since the information to transmit consists the content of variable transmitted is small. What it is relevant is the latency between packets, even when the adaptation of the user interface is not intended to be changed often, the estimator must have enough time for computing the output.

Future studies can evaluate the benefits of using different standard wireless communication protocols as Bluetooth, Bluetooth Low energy, or Wi-Fi for transmitting information. Some application layer protocols can also be used to transmit information as CoAP (Constrained Application Protocol) or EXLAP (Extensible Lightweight Asynchronous Protocol) which is explained below. Some variables to consider in the analysis may be energy consumption, bandwidth, latency and availability among different devices.

Communication with the vehicle

The communication between the modules that integrate a vehicle is mainly managed through a bus called Controlled Area Network bus (CAN bus). This communication standard was patented by BOSCH in 1982 and it was initially designed to allow the exchange information between micro-controllers. Given its high reliability, it was adopted by vehicle manufacturers for connecting the different Electronic Control Units (ECUs) of the vehicle (Boys (2004) and UPV (2012)). The electronic architecture of a vehicle is quite complex and is composed by several CAN buses, the distribution of messages between the different CAN buses is done by a central unit called Gateway.

CAN bus is an asynchronous protocol with the following features:

- Implements Carrier Sense Multiple Access/Collision Detection (CSMA /CD).
- Is Multicast, so every ECU can listen and send messages.
- Is based on broadcasting messages, any ECU connected to the bus listens messages transmitted.
- Integrates error correction detection mechanisms.
- Implements a priority scheme to send relevant messages faster than others.
- Uses as physical medium a pair of twisted cables to reduce interference, it supports a speed of 1 Mbps at a maximum length of 40 m. In vehicles systems there are used speeds of 125 Kbps or 500 Kbps.

On top of the CAN bus, Volkswagen implements a proprietary protocol called Bedien- und AnzeigeProtokoll (BAP). This protocol sends multiplexed messages inside CAN frames. BAP offers a bidirectional communication which permits to request information to specific controllers. Four main types of BAP messages are defined: Get (request information), SetGet (Request an update of a value), Status (response to GET message) and HeartBeat (Broadcast notification of information).

The access to the information available in the CAN bus is not complicated to achieve. Several electronic components have been designed for this purpose and a large amount of libraries exists which ease the monitoring and update of some variables. During the development of the project, two approaches were implemented. The first one consisted in the integration of a CAN-to-WiFi module, this module was connected physically to two CAN channels and the data was made available trough a socket over WiFi.

The second approaches consisted in the installation of a Linux machine in the vehicle, this machine had a complete access to all the buses thanks to the integration of CAN to USB adapters. The access from a client application was possible thanks to a service available for Linux called “SocketCan”, this service allowed the subscription of external devices to CAN variables trough a TCP port. In a similar manner, the Linux machine had also a WiFi interface and the access was also possible trough a wireless connectivity.

With the growth of third-party developers, it became necessary to provide a secure protocol that give access to certain information in the vehicle. Some variables transmitted trough the CAN bus are very sensitive since they are associated to engine, breaks, direction or air bags information. An incorrect use of CAN buses as the overwriting of messages or an excessive transmission of messages causing congestion could jeopardize safety. As a solution, the VW consortium introduced a new communication protocol called "Extensible Lightweight Asynchronous Protocol" (EXLAP).

EXLAP is a communication protocol that allows the access to certain CAN bus information from an external device through a USB cable, Bluetooth or Wi-Fi. It is added a security layer in which the client application must have a certificate for login into the service. The service is based on subscriptions, the client side request to be notified about the changes of a variable in a given period, or it can also send requests to execute certain actions. EXLAP is more oriented to be used for interacting with infotainment systems, therefore a list of commands related to navigation, media and radio functionalities are available. Several "Application Programming Interfaces" (APIs) have been developed for Android, IOS and Java which ease implementing this protocol in external applications.

3.2 Data Mining Tools

A large amount of variables that may define the driving complexity have been exposed in Section 2.2. The degree of impact of each one is unknown and it can be expected that the driving complexity is unlikely to be expressed as a single linear combination of these variables. At contrary, the outcome is expected to be the result of complex relations between the variables, e.g, an increment of the speed in a motorway may not represents the same as in a rural road, or a high rotation of the steering wheel angle is expected to have a different impact depending on the speed. A generic model that represents the Driving Complexity is presented in Equation 3.1, here each variable $(x, y, z, ..)$ affects the driving complexity in a different degree and its effect is conditioned by other variables.

$$DrivingComplexity = f_x(y, z, ...) + f_y(x, z, ..) + f_z(x, y, ...) + ... \tag{3.1}$$

As an alternative to a mathematical model, many statistical analyses employ data mining techniques to reveal the relationships between the variables. This approach is very useful when the complexity of these relations is large or when the model is expected to be adaptable in case a future incorporation of a more data. Moreover, machine learning methods can be used for creating models capable of fitting an unknown outcome given a set of inputs.

The implementation of machine learning methods with the goal of estimating driving complexity has been already studied. Some of the most notable researches made use of neural networks in order to construct a model for workload categorization (Yoo et al., 2006) and (Hoogendoorn and Van Arem, 2013). They identified that some intrinsic relations in the workload characterization are very difficult to model and this type of machine learning method offers an alternative to find suitable models. Overall, the models created achieved a good accuracy and predictability.

In another study, variables related to the vehicle such as the average speed, the standard deviation distance of the lane and the longitudinal acceleration were used for building a workload estimator (Zhang and Owechko, 2004). Participants were asked to perform secondary tasks while driving a vehicle simulator and data collected was used for training a machine learning methods based on decision trees. The final accuracy of the model also showed good results given the sample of participants used.

Regarding the software tools that can be used for analysing data and constructing models using machine learning methods, several software tools are available for Windows, Linux, and Mac OS X operating systems. A comparison of the most known free tool software exposed in ((Jović et al., 2014) and (Borges et al., 2013)) are: RapidMiner (Hofmann and Klinkenberg, 2013), R (R Development Core Team, 2008), Weka (Hall et al., 2009), Knime (Berthold et al., 2007) and Orange (Demšar et al., 2013).

In terms of more available features and capabilities, RapidMiner, R, Weka, and Knime integrate most of the tools expected when performing data mining analysis and therefore can be recommended for most of the tasks needed in this study. In relation to performance and accuracy of the algorithm implemented more debate exists. A comparison of Weka, Knime, Orange and RapidMiner in terms of average accuracy for 8 different datasets and 4 different machine learning methods is presented in (Borges et al., 2013).

Table 3.1 summarize these accuracy results, the average of all the machine learning method accuracy is shown in the last column where Knime reached the highest result. This value does not show which is the best performing method since the outcome depends on the dataset type as well as the method applied, e.g., Weka is much better than Knime when a Bayesian Classifier is used but worst when decision trees are selected. Knime also incorporates several Weka packages which implement the algorithms developed in Weka (plugins).

Table 3.1: Comparison of Data Mining software for different methods (average accuracy for different datasets).

Tool	Decision Trees	ANN	Rules	Clustering	Bayesian Classifier	SVM	Average
Weka	58,96	83,23	85,89	83,09	85,11	82,4	79,8
Knime	83,06	84,38	80,17	83,29	76,41	81,06	81,4
Orange	81,21	-	80,67	81,43	82,05	80,53	81,2
RapidMiner	82,15	80,93	84,06	80,49	83,48	54,87	77,7

Source: Borges et al. (2013).

Another recommended software that implements machine learning methods is Encog (Heaton (2015)). Encog is a framework that support a large amount of classification and regression algorithms. In comparison to previous tools, this software does not offer a block oriented GUI, the programming can be done using their GUI or directly writing code lines. The framework has been implemented in several programming language such as Java, .NET and C++.

In this first part of the methodology, and with the objective of selecting the most appropriate tool for the study, Encog and Knime are compared in terms of prediction accuracy, training time and overall capabilities.

3.3 Machine Learning: Theoretical Evaluation

The selection of the most optimal data mining tool and technique depends on the dataset and the purpose of the application. This section compares different machine learning methods and tools in terms of two metrics: the computational training time needed for creating the model and its final accuracy once trained.

Machines learning methods are compared and evaluated using two different platforms: Knime data mining software and Encog framework. The first one is a data mining and analyser software which has a rich graphical environment and set of tool for manipulating data and performing data mining analysis. The second one provides a framework in Java that allows defining and training machine learning methods, this framework was developed by the Heaton Research. The last available versions of each platform at the moment of the test were used: Knime 2.12 and Encog Java library 3.3.

In order to compare different machine learning methods, a generic dataset expected for the proposed system, was initially created. Under this approach, several characteristics can be modified in order to evaluate the machine learning performance under different scenarios that may not be very frequent or easy to record in real driving scenarios. Moreover, a dataset containing every possible combination of inputs (incoming vectors) is difficult to acquire in a real driving scenarios. Therein lies the importance of using simulated datasets for understanding the behaviour of the method under certain circumstances. Some of the characteristics that can be changed are the number of parameters, structure, size and noise present in the dataset. Data was structured in a comma delimited file (.csv) where each row defined the circumstances at a specific time (case). The combination of inputs that define the scenario is named "incoming vector", and the output is called "output vector".

The parameters configured in each training set are as follows:

- Incoming Variables: Indicates the number of input variables to the system. A unique set of incoming variable values is defined as an "incoming vector". The same incoming vector could appear several times in a dataset and be associated to different outputs; in this case the uncertainty increases and wrong values are categorized as noise.

- States per variable: In this evaluation, variables are quantified instead of being expressed as a continuous value. The number of states provides information about the total amount of possible values, therefore it can be taken as a reference of precision. A high precision, closer to the continuous value, has the counterpart of increasing the complexity of the data. On the other hand, few states can lead to less precision. In these initial tests, with the only purpose of comparing the data mining tools, the variables are defined with 3 states.

- Training set/ Validation set sizes: In the same way as it is used a trained set for training the method, it is needed a validation set for evaluating it the machine learning performance performance. Usually, the complete dataset is divided in two groups, a big percentage of the dataset is used for training and the other is left for validation.

- Output Noise: It is not expected that each incoming array is always associated to the same output. In real systems, unexpected outputs must be taken into account. The machine learning method should be capable of weighting repeated incoming vectors and adapt the response depending on the uncertainty of the output. In this test, for validating outputs, any prediction equal to the mode/s of the possible outputs for the incoming vector is considered correct. Therefore, the level of noise is computed as the percentage of incorrect predictions that differ from the mode/s. Below, some noise calculation examples are presented:
  - Outputs for same incoming vector: [0,0,0,0], Noise = 0%
  - Outputs for same incoming vector: [0,0,0,1], Noise = 25%
  - Outputs for same incoming vector: [0,0,1,1], Noise = 0%
  - Outputs for same incoming vector: [0,0,1,1,0.5], Noise = 20%
  - Outputs for same incoming vector: [0,0.5,1,1,1,0], Noise = 50%

*The total percentage of noise in the dataset is computed as the average of noises resulting for each set of incoming vectors. In example, given a dataset composed that contains the outputs previously presented (23 vectors), the average noise resulting will be 21.7%.

The first phase of the test consisted in the creation of datasets. In this regards, two classes of datasets with different levels of noise were defined: dataset class A with 0% level of noise, and dataset class B with a 30% of noise approximately. For each class, 4 subclasses with a different number of input variables (3,4,5,6) were created. Table 3.2 shows the different dataset available for evaluating the performance of machine learning methods. The third column indicates the total amount of possible combinations that a dataset can have (input states per variables = 3). The fourth column shows the number of rows in the datasets, each combination occurs 10 times in each dataset, in case of error the same combination may have a different outputs associated.

Table 3.2: Datasets used in the simulations.

Case	Variables	Combinations	Rows	Noise Level
A3	3	27	270	0
B3	3	27	270	≈30
A4	4	81	810	0
B4	4	81	810	≈30
A5	5	243	2430	0
B5	5	243	2430	≈30
A6	6	729	7290	0
B6	6	729	7290	≈30

The second phase consisted in the training and evaluation of the different machine learning methods in terms of training time and accuracy. The training and evaluation was repeated three times in order to reduce the effect of randomly initiated variables that some methods apply by default.

The following Machine learning methods were tested, evaluated and compared using simulated datasets in two platforms, Encog and Knime:

- Artificial Neural Networks (ANN).
  - Multi-Layer Perceptron (MLP): The performance of this type of network depends on its topology, therefore several topologies were created and tested. The complexity of the network was increased adding more hidden layers or nodes per layer. The final goal was to find the more accurate network with the simpler topology.
  - Probabilistic Neural Networks (PNN): This kind of networks have a fixed topology consisting always of 4 layers. The increment in the number of nodes present in internal layers is automatically performed by the algorithm.
- Support Vector Machines (SVM): The method make use of Kernels to project point into a higher dimensional space and overcome computational limitations.
- Hyper NeuroEvolution of Augmenting Topologies (HyperNEAT): This machine learning method solves the problem of finding the most suitable Neural Network topology.

It employs genetic algorithms for adapting the topology in term of prediction accuracy. This method is not available in Knime and therefore it is only be tested in Encog.

- Fuzzy Rule Leaner: Creates a set of rules for classifying data. Encog does not support this method and it is only be tested in Knime.

- Decision Tree Learner: Creates set of classification rules based on statistical information. Encog does not support this method and it is only be tested in Knime.

Test were executed on a computer with the following features:

- Machine: EliteBook 8470p.
- CPU: Intel(R) Core^{TM} i5-3320M @2.6GHz.
- RAM: 8,00GB.
- Operative System: Windows 7 Enterprise 64 bits.
- Java Version: 1.8.0_60 (Java HotSpot^{TM} 64-Bit Server).

The following subsections expose the analysis of different machine learning methods using the two data mining software selected: Encog and Knime.

3.3.1 Encog

The machine learning methods compared using the Encog Framework were: MLP, PNN, SVM and HyperNEAT. For this purpose, the simulated dataset was initially transformed to a bin format, this procedure is necessary since large datasets could overload memory available to the Java Heap and result in running out of memory. Following, subsections present the performance results obtained of each method tested.

MLP

In a MLP, the following parameters are configured:

- Network Topology: Number of hidden layers, number of nodes per hidden layer and presence of bias in layers. A complex topology does not always guarantee good accuracy since these networks are susceptible to over-fitting.

- Activation Function: Define the nodes response to incoming values; activation functions tested are as follows: BiPolar, BipolarSteepnedSigmoid, ClippedLinear, Gaussian, SteepenedSigmoid, Step, Ramp, Linear, Elliott, ElliottSymmetric, Log, sin, Sigmoid, Tanh.

- Propagation type: Defines the technique used for training: RPROP+, RPROP-, iRPROP+, iRPROP, New RPROP without weight back tracking and ARPROP (Nonlinear Jacobi RPROP).

- Regression or Classification: Boolean that defines the behaviour of the machine learning method to compute a regression or classification of data.

Given the large amount of parameters and number of possible configurations, the first phase consisted in evaluating the performance of different activation functions letting fixed the propagation type to *iRPROP+* (recommended according to Section 2.4.2), and iterating among different neural network topologies. The number training iterations is limited to 200, the expected error to 0% and the most complex topology allowed was limited to 10 hidden layers with 40 nodes per layer. In this initial phase, the dataset used was the A3 (0% of noise).

The initial network topology consisted of 1 layer and 2 nodes, this topology was increased in complexity adding 2 nodes to each hidden layer up to 40, and then adding a new hidden layer until the number of layers reached 10 (all layers included bias connections). The algorithm used for increasing the complexity is presented in Listing 3.1.

Listing 3.1: Algorithm for increasing network complexity.

```
int error = 1;
int layers = 1;
int nodes_per_layer = 0;
while(error>0 && layers<=10){
    if(nodes_per_layer<40){
        nodes_per_layer = nodes_per_layer + 2;
    }else {
        layers++;
        nodes_per_layer = 2;
    }
    error = check_error();
}
```

Performance results (average accuracy and training time) for each activation function are summarized in Table 3.3. The best activation functions in terms of average prediction accuracy (more than 97%) are highlighted in green. From these highlighted cases, the best performing in terms of average training time (less than 400 ms) are: Log for regression, Sigmoid for classification, Tanh for regression, and Elliot Symmetric for both cases. It is also seen how classification usually outperforms regression in terms of average accuracy for almost any activation function. Overall, the best performing activation function is Elliot Symmetric for a classification scheme.

The best performing activation functions in terms of average accuracy (highlighted in Table 3.3) are tested with a new dataset. In this case, the dataset with 30% of noise (Dataset B3) is used for training. Summary of results are presented in Table 3.4. From the networks capable of achieving 100% of accuracy (Highlighted in green), those that offer the lower training time are selected: Elliot and Elliot Symmetric. These networks also seem to provide a proper consistency of results, which is understood as the reliability of a network to behave similarly over different trainings with certain level of noise. It is also seen how a classification scheme outperforms a regression scheme in terms of accuracy and training time.

Using a Classification scheme and the best performing activation functions, Elliot and Elliot Symmetric, different back propagation techniques are evaluated. Results are presented in Table 3.5; for every dataset and activation function the best performance is always obtained using the back propagation algorithm iRPROP+. This result matches with other researches as (Igel and Hüsken (2003)) where different back propagation algorithms were compared. It is also detected that Elliot Symmetric outperforms Elliot when dataset B3 is used (noise added). As a conclusion, the preferred MLP training configuration for the defined scenario uses an Elliot Symmetric activation function, a classification scheme and an iRPROP+ propagation algorithm.

Table 3.3: MLP performance analysis of activation functions, 0% noise, (Encog).

Activation Function	Regression or Classification	Avg. Accuracy [%]	Avg. Training Time [ms]
BiPolar	Regression	43 ± 2	648 ± 26
BiPolar	Classification	50 ± 4	657 ± 79
Bipolar Steepened Sigmoid	Regression	78 ± 11	1051 ± 136
Bipolar Steepened Sigmoid	Classification	86 ± 8	1036 ± 37
Clipped Linear	Regression	91 ± 10	544 ± 207
Clipped Linear	Classification	85 ± 10	679 ± 53
Gaussian	Regression	56 ± 2	1419 ± 11
Gaussian	Classification	44 ± 6	1491 ± 47
Steepened Sigmoid	Regression	56 ± 2	1457 ± 120
Steepened Sigmoid	Classification	61 ± 8	1438 ± 78
Step	Regression	45 ± 6	713 ± 63
Step	Classification	73 ± 2	678 ± 48
Ramp	Regression	86 ± 8	664 ± 107
Ramp	Classification	95 ± 6	558 ± 128
Linear	Regression	32 ± 5	587 ± 21
Linear	Classification	44 ± 0	615 ± 31
Elliot	Regression	97 ± 2	742 ± 334
Elliot	Classification	100 ± 0	480 ± 98
Elliott Symmetric	Regression	100 ± 0	386 ± 64
Elliott Symmetric	Classification	100 ± 0	183 ± 143
Log	Regression	100 ± 0	317 ± 120
Log	Classification	100 ± 0	533 ± 167
Sin	Regression	100 ± 0	471 ± 132
Sin	Classification	100 ± 0	503 ± 12
Sigmoid	Regression	100 ± 0	472 ± 148
Sigmoid	Classification	99 ± 2	394 ± 75
Tanh	Regression	100 ± 0	339 ± 109
Tanh	Classification	100 ± 0	557 ± 93

Table 3.4: MLP performance analysis of activation functions, 30% noise, (Encog).

Activation Function	Regression or Classification	Avg. Accuracy [%]	Avg. Training Time [ms]
Elliot	Regression	88 ± 0	978 ± 272
Elliot	Classification	100 ± 0	474 ± 158
Elliott Symmetric	Regression	87 ± 2	992 ± 287
Elliott Symmetric	Classification	100 ± 0	515 ± 168
Log	Regression	88 ± 0	1164 ± 326
Log	Classification	100 ± 0	568 ± 113
Sin	Regression	85 ± 0	2005 ± 559
Sin	Classification	100 ± 0	544 ± 88
Sigmoid	Regression	84 ± 2	1603 ± 527
Sigmoid	Classification	93 ± 6	1145 ± 570
Tanh	Regression	85 ± 0	1865 ± 471
Tanh	Classification	100 ± 0	698 ± 155

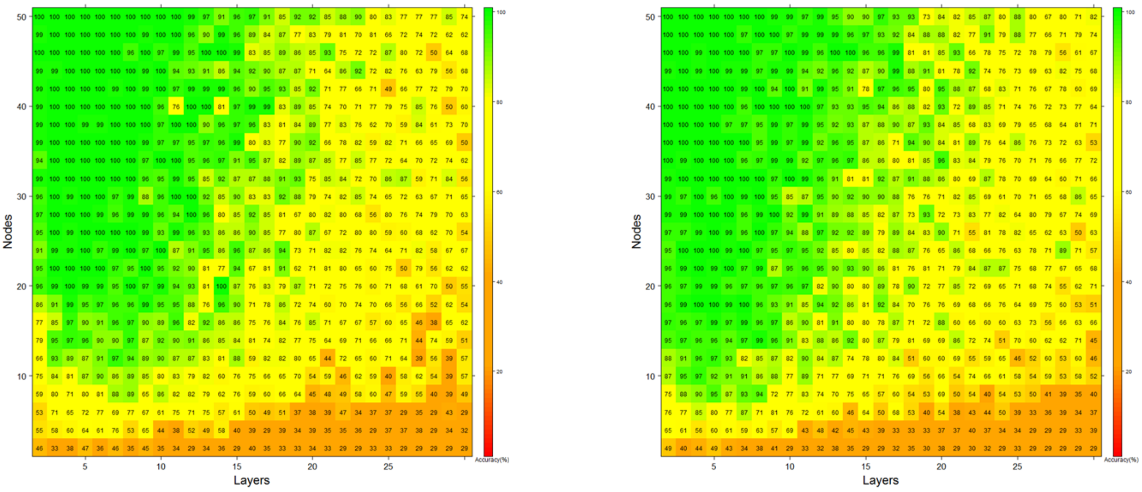
Table 3.5: MLP performance analysis regarding Back Propagation techniques (Encog).

Dataset	Activation Function	Propagation	Avg. Accuracy [%]	Avg. Training Time [ms]
A3	Elliot	RPROP+	96 ± 0	974 ± 255
A3	Elliot	RPROP-	95 ± 2	1013 ± 263
A3	Elliot	iRPROP+	100 ± 0	374 ± 63
A3	Elliot	iRPROP-	92 ± 0	851 ± 425
A3	Elliot	ARPROP	58 ± 17	981 ± 234
A3	ElliottSymmetric	RPROP+	100 ± 0	504 ± 191
A3	ElliottSymmetric	RPROP-	100 ± 0	577 ± 22
A3	ElliottSymmetric	iRPROP+	100 ± 0	445 ± 94
A3	ElliottSymmetric	iRPROP-	100 ± 0	500 ± 159
A3	ElliottSymmetric	ARPROP	99 ± 2	654 ± 76
B3	Elliot	RPROP+	92 ± 0	958 ± 256
B3	Elliot	RPROP-	87 ± 2	1001 ± 293
B3	Elliot	iRPROP+	100 ± 0	623 ± 235
B3	Elliot	iRPROP-	86 ± 2	1011 ± 163
B3	Elliot	ARPROP	45 ± 9	1016 ± 222
B3	ElliottSymmetric	RPROP+	100 ± 0	643 ± 152
B3	ElliottSymmetric	RPROP-	100 ± 0	724 ± 8
B3	ElliottSymmetric	iRPROP+	100 ± 0	525 ± 163
B3	ElliottSymmetric	iRPROP-	100 ± 0	576 ± 128
B3	ElliottSymmetric	ARPROP	81 ± 21	808 ± 299

Finally, the last parameter to evaluate is the topology of the network. In this regards, different topologies were tested for the same dataset. The process was repeated three times and the average result was the value used as reference. The topology was modified iterating the number of hidden layers from 1 to 30 and nodes per hidden layer from 2 to 40 in steps of two. The MLP configuration used consisted of: Elliot Symmetric activation function, classification scheme and an iRPROP+ propagation algorithm.

Figure 3.3 shows the accuracy performance depending on the topology, here the colour green represents a high accuracy and colour red a lower one. For both datasets, it is appreciated how complex topologies do not necessarily have better accuracies, which means that the topology configuration must be careful chosen. The optimal network that provides an adequate accuracy and have a simpler topology is the one that have a configuration of 7 layers and around 25 nodes per layer.

Figure 3.3: MLP accuracy depending the topology.



(a) MLP Accuracy depending on topology (Dataset A)

(b) MLP Accuracy depending on topology (Dataset B)

PNN

PNN has a simple topology consisting in a feed forward neural network with a fixed number of layers: Input Layer, Pattern/Summation Layers and Output Layer. In Encog, it is possible to choose between regression or classification training and also select the Kernel which can be Gaussian or Reciprocal.

Table 3.6 shows the performance results after training a PNN network changing the Kernel and using datasets A3 (without noise) and dataset B3 (around 30% noise). The classification scheme outperforms regression scheme when noise is present. In relation to the Kernel, the reciprocal Kernel requires the lowest training time among all.

Table 3.6: PNN performance regarding Kernel Type (Encog).

Dataset	Kernel	Regression or Classification	Avg. Accuracy [%]	Avg. Training Time [ms]
A3	Gaussian	Regression	100 ± 0	21 ± 1
A3	Gaussian	Classification	100 ± 0	30 ± 10
A3	Reciprocal	Regression	100 ± 0	10 ± 0
A3	Reciprocal	Classification	100 ± 0	30 ± 1
B3	Gaussian	Regression	70 ± 0	267 ± 6
B3	Gaussian	Classification	100 ± 0	271 ± 1
B3	Reciprocal	Regression	70 ± 0	50 ± 1
B3	Reciprocal	Classification	100 ± 0	44 ± 6

This kind of network converges very fast to a minimum error, for the datasets evaluated A3 and B3 it is only needed one iteration to create a model that achieves a 100% of accuracy; nevertheless the processing time of each iteration is larger in comparison to MLP. Among the different configuration, a PNN with a reciprocal Kernel and a classification scheme seems the most suitable method.

SVM

In Encog, the SVM training type can be changed. For training, the number of iteration was set to a maximum of 100 but after approximately 10 iterations the value did not change significantly. Table 3.3.1 shows the performance results for different SVM types. Among the method tested, the most suitable in terms of accuracy and training time was "Support Vector" for a classification scheme, it offers a 100% of accuracy for both datasets A3 and B3, and also has the lowest training time.

Table 3.7: SVM performance regarding SVM Type (Encog).

Dataset	SVM Type	Scheme	Avg. Accuracy [%]	Avg. Training Time [ms]
A3	EpsilonSupportVector	Regression	100 ± 0	200 ± 10
A3	NewSupportVector	Regression	100 ± 0	317 ± 6
A3	SupportVector	Classification	100 ± 0	77 ± 15
A3	NewSupportVector	Classification	100 ± 0	381 ± 125
B3	EpsilonSupportVector	Regression	74 ± 0	687 ± 126
B3	NewSupportVector	Regression	88 ± 6	710 ± 87
B3	SupportVector	Classification	100 ± 0	641 ± 55
B3	NewSupportVector	Classification	100 ± 0	787 ± 127

The training time was observed to be considerably affected by the level of noise. This is as a consequence of the intrinsic algorithm operation, SVM convergence time is affected by dispersion of points since the hyper plane that separate the classes is more difficult to find.

Hyper NeuroEvolution of Augmenting Topologies (HyperNEAT)

This kind of networks deal with the problem of identifying the best MLP topology; for this purpose a genetic algorithm is used for finding the best topology. Drawbacks of this method are the large amount of parameters related to the genetic algorithm that needs to be adjusted, and the convergence time that could be also very slow. Parameters that can be adjusted in this method are the population size, the probability of crossover and probability of mutation, all of them very related to the convergence speed and accuracy (Assistant (2014)). Mutations create new species in the population which are useful for exploring new areas and avoid falling in local minimums. On the other hand, crossovers help to locate local minimums mixing optimal genomes.

Table 3.8 shows 10 different configurations evaluated for available parameters. In each configuration some of the parameters are modified. These values indicate the probability of an event to happen in each iteration (mutation or crossover).

Initially, the HyperNEAT method was evaluated using the default configuration, the training set A, and changing the size of the population (50,100,150), see the first part of Table 3.9. From these results, the regression scheme is observed to have a better accuracy performance than the classification scheme. With regards to the population size, an amount equal to 50 seems to be enough since the accuracy does not increase much when the population is bigger, but the training time is clearly affected. Using a population size of 50 and a regression scheme, the performance is evaluated for the different evolutionary configurations defined.

From these results, the configuration "I" (Highlighted in green) is the best performing

Table 3.8: Genetic Algorithm parameters configuration.

Evolutionary Parameter	Operator Probability				
	Crossover	Weight Mutation	Add Node Mutation	Add Link Mutation	Remove Link Mutation
Default	0,5	0,494	0,0005	0,01	0
A	0,6	0,394	0,0005	0,01	0
B	0,7	0,294	0,0005	0,01	0
C	0,4	0,594	0,0005	0,01	0
D	0,3	0,694	0,0005	0,01	0
E	0,3	0,64	0,005	0,05	0,01
F	0,3	0,6	0,02	0,07	0,01
G	0,3	0,5	0,05	0,1	0,05
H	0,25	0,4	0,1	0,2	0,05
I	0,25	0,35	0,1	0,25	0,05

one when it respect to accuracy. Nevertheless, this accuracy value is small, 44%, and the training time raised up to 23 seconds. Even when the performance could be improved selecting more suitable parameters, this method is much slower for the same dataset than other previously evaluated such as SVM or PNN. Moreover, the selection process of optimal evolutionary parameters compromises a future addition of more input variable. As a consequence, this kind of networks is discouraged given the purpose of this study and they will not be taken into account in the final comparison.

Table 3.9: HyperNeat performance results.

Scheme	Population	Evolutionary Parameters	Avg. Accuracy [%]	Avg. Training Time [ms]
Regression	50	Default	37 ± 4	20379 ± 559
Regression	100	Default	39 ± 4	37844 ± 1400
Regression	150	Default	39 ± 2	60390 ± 2496
Classification	50	Default	29 ± 0	22450 ± 405
Classification	100	Default	30 ± 2	44826 ± 1174
Classification	150	Default	29 ± 0	68670 ± 981
Regression	50	A	36 ± 2	18678 ± 2066
Regression	50	B	36 ± 2	19850 ± 2763
Regression	50	C	37 ± 0	19581 ± 1399
Regression	50	D	37 ± 4	19726 ± 1717
Regression	50	E	33 ± 0	15622 ± 13497
Regression	50	F	34 ± 2	22963 ± 744
Regression	50	G	37 ± 4	22761 ± 389
Regression	50	H	41 ± 6	22781 ± 724
Regression	50	I	44 ± 4	23298 ± 428

Overall comparison (Encog)

Once different methods were tested and optimized independently, they were compared in terms of training time, accuracy and final model size. The training of each method was limited to 200 iterations, the datasets used were presented in Table 3.2 and the optimization parameters for each case are presented below:

- MLP: Elliot Symmetric activation function, Classification scheme, iRPROP+ propagation algorithm, maximum of 10 hidden layers and maximum of 40 nodes per hidden layer.
- PNN: Classification scheme and Kernel: Reciprocal.

- SVM: Support Vector Classification type and RBF Kernel.

Figure 3.4 shows the accuracy performance of the different methods evaluated using Encog library. Methods SVM and PNN were both capable of generating models that achieved a 100% of accuracy for every dataset, MLP on the other hand did not reach 100% of accuracy when datasets had more than 5 inputs; this is attributed to the limitation on network complexity topology and number of iterations.

In regards of the training time (Figure 3.5), the method with the lowest training time was PNN, this method was capable of reaching 100% of accuracy in just one iteration. A great difference is observed when the dataset size is higher e.g.,B6, PNN was capable of achieving 100% of accuracy and training time was 50 seconds while SVM training time took 201 seconds; MLP achieved a 43% of accuracy and the training time was 54 seconds. The noise level clearly affects the training time and accuracy for every case.

Figure 3.4: Accuracy obtained by different methods (Encog).

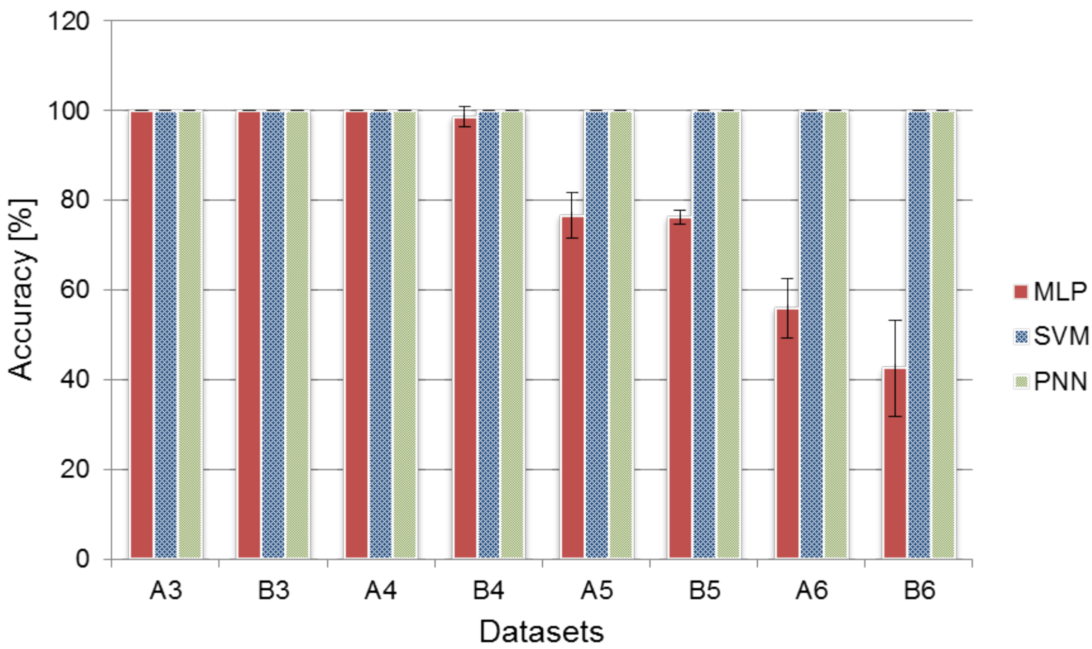


Figure 3.5 seems to shows an exponential relation between training time and dataset size. This relation could be associated to the number of inputs variables or by the size of the dataset independently the number of inputs. Two tests are performed in order to compare the effect of each parameter. In the first one, the number of inputs is increased while the dataset is kept to a fixed size; results show a logarithmic (base 10) relation with a fitness of R^2=0.98 between number of inputs and training time.

In the second test, the dataset size was increased while the number of inputs was kept to a fixed value of 3. Results presented in Figure 3.6 show that the training time of the SVM machine method is significant lower than the training time of the PNN method for big datasets and fixed number of inputs. In conclusion, SVM training time is very affected by the dataset diversity (number of inputs); on the other hand PNN is more susceptible to the dataset size (number of rows).

Figure 3.5: Training time consumed by different methods (Encog).

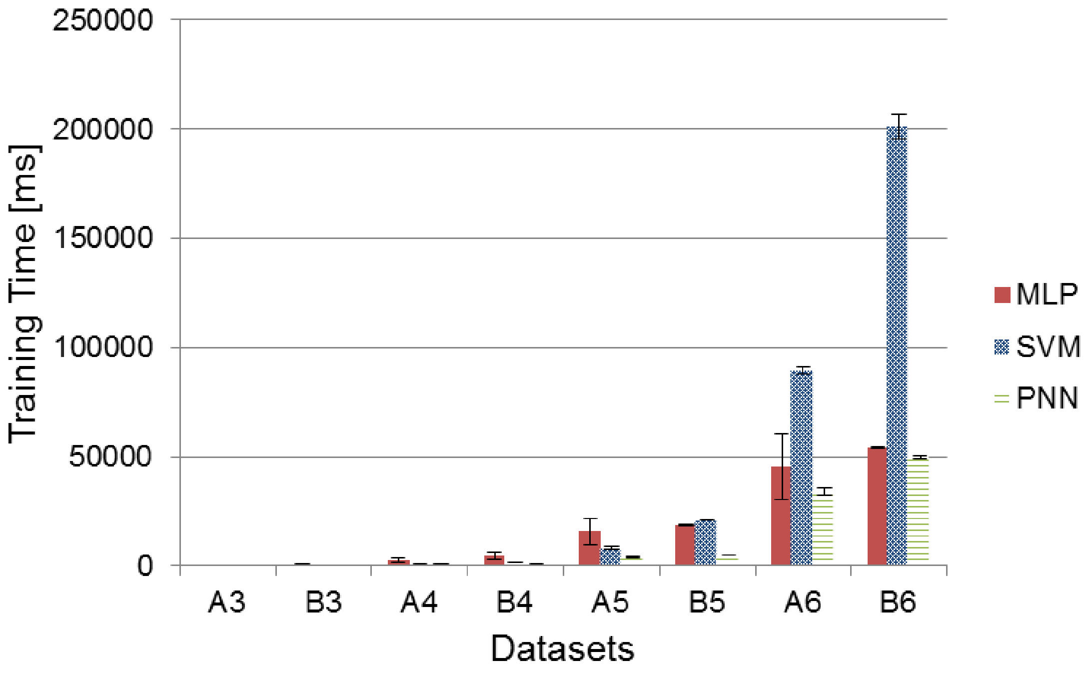
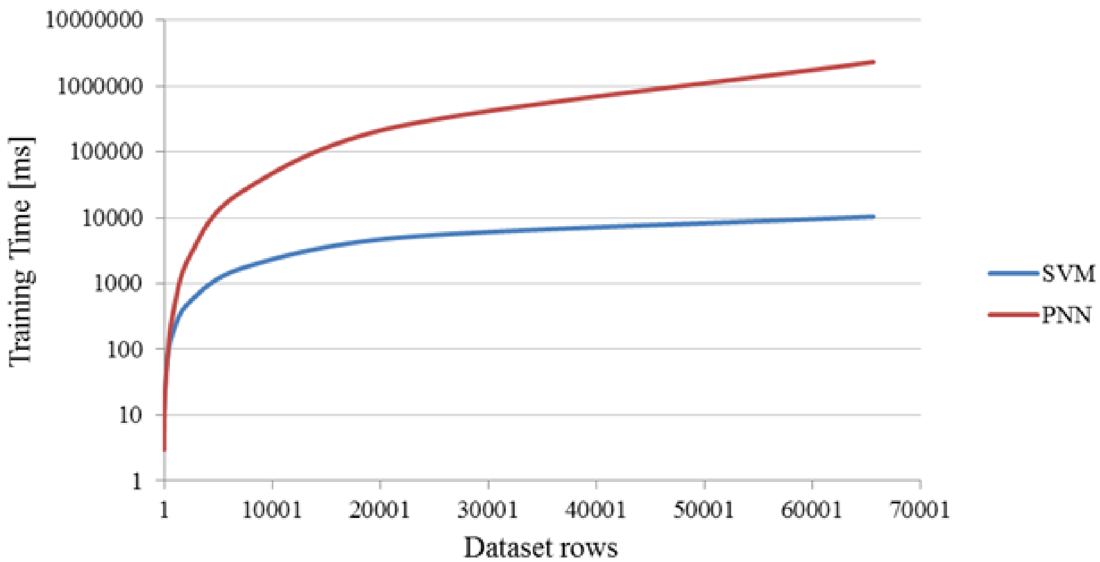


Figure 3.6: Training time for the same number of incoming variables (3) and different dataset sizes (Encog).



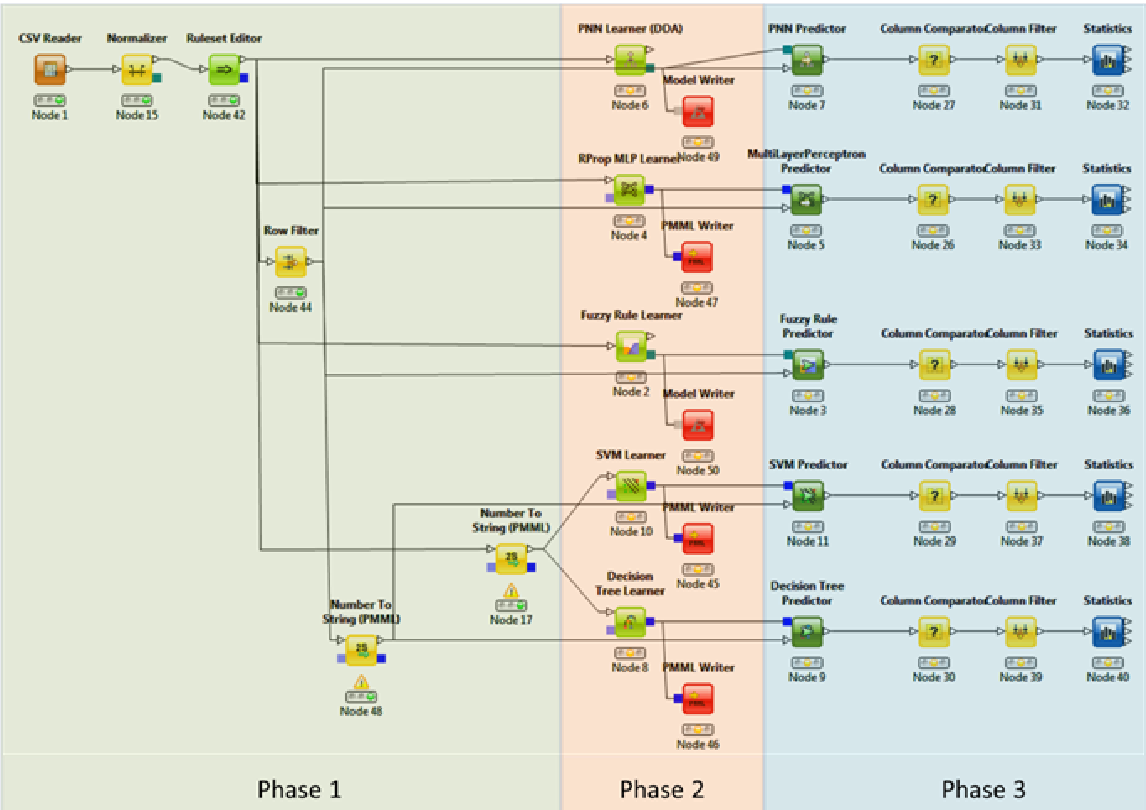
The last parameters evaluated was the needed storage size in memory for saving the model, this value is important in case the model needs to be updated and kept in devices with small storage capacities. The size of the model depends on the dataset used during training. For dataset B6, the following model sizes were obtained: MLP reached 161KB, SVM 316KB and PNN 136KB. Overall, PNN created the lighter model for every dataset tested which make it the most suitable to be saved in devices with low storage capacity.

To summarize Encog evaluation, SVM and PNN are the most suitable methods among the different tested for the employed datasets. Some overall conclusions are than, PNN outperforms SVM when data is very diverse while SVM outperform PNN if the dataset is big but not so diverse.

3.3.2 Knime

Software Knime v2.12 was used to create a generic workflow where different machine learning methods were compared: PNN, MLP, SVM, Fuzzy Logic and Decision Trees. The designed workflow consisted of three main phases presented in Figure 3.7, the initial phase executed a reading and normalizing of the csv file that contained the training set, the second phase was the learning phase in which machine learning methods are trained, and the final phase consisted in the calculation of values by the trained model and computation of performance statistics.

Figure 3.7: Workflow in Knime (testing workbench).



Following the same approach undertaken during Encog tests, each machine learning method was configured depending on certain parameters and were compared in terms of accuracy and training time. The configuration parameters, which can be easily updated in each training block (Learners shown in Figure 3.7), for each method are as follows:

- RPROP MLP: Maximum number of training iterations, number of Hidden Layers, number of hidden Nodes per hidden layer.
- PNN: Maximum number of training iterations, theta minus, theta plus,
- Fuzzy Rule: Maximum number of training iterations, fuzzy normalization (Min/Max, Product, Lukasiewicz, Yager 0.5), shrink function (Volume Border Based, Voluma Anchor Based, Volume Rule Based)
- SVM: Kernel (Polynomial, Hyper tangent, RBF)
- Decision Trees: quality measure, Gini or Gain index, pruning method (None or MDL).

MLP

The performance of the MLP network was evaluated in terms of accuracy and training time for different topologies. The number of iterations during training was fixed to a maximum of 200 and the datasets tested were Dataset A3 (0% of noise) and Dataset B3 (30% of Noise). In both cases a 100% of accuracy was achieved, for Dataset A3 the training time raised to 188ms while for Dataset B3 the training time was 251ms.

Figure 3.8 shows performance results in terms of accuracy for different topologies when Dataset A is used. Network complexity was increased adding 2 nodes to each hidden layer up to 50, and then adding a new hidden layer up to 27. The pattern obtained is very similar to the obtained when using Encog; in this case the optimal network topology is achieved using 5 layers and around 15 nodes. The conclusion is the same previously exposed, a complex topology does not guarantee a good accuracy if the number of iteration is limited. And in case that the iteration limit is removed, the training time will be clearly affected.

PNN

Knime allows to configure several PNN parameters such as the number of training iterations, which was limited to 100 and the values of theta minus and theta plus. Theta minus defines the upper boundaries of activation for conflicting rules and theta plus defines the lower boundary of activation for non-conflicting rules.

The PNN performance was evaluated for datasets A3 and B3, and for different values of theta minus and theta plus (Table 3.10). Results when using dataset B3 are not presented here since the accuracy obtained was always 70%. In the Appendix Section B (Table B.1) is presented this information. It is seen how the higher accuracies are obtained when the value of theta minus is equal to 0.1 (highlighted in green). With regards to training time, the value obtained was similar for every case (between 21 and 26 milliseconds).

Figure 3.8: MLP accuracy depending the topology in Knime (Dataset A).

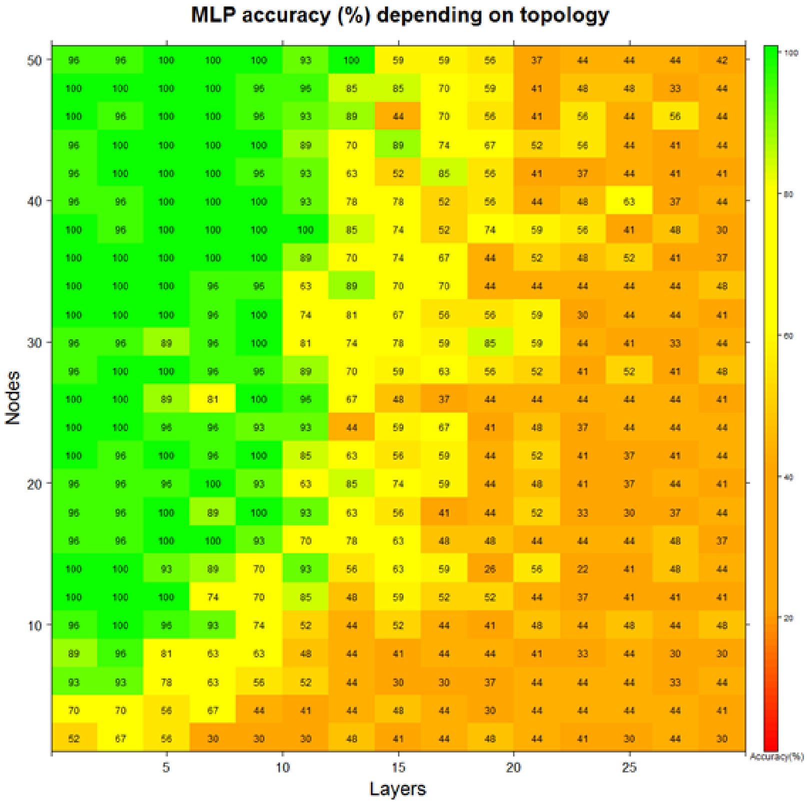


Table 3.10: Accuracy and training time for PNN depending on theta values, Dataset A3 (Knime).

theta plus	theta minus	Avg. Accuracy [%]	Avg. Training Time [ms]
0,1	0,3	89 ± 0	20 ± 9
0,1	0,5	56 ± 0	21 ± 10
0,1	0,7	41 ± 0	15 ± 1
0,1	0,9	30 ± 0	16 ± 1
0,2	0,1	100 ± 0	21 ± 9
0,2	0,3	93 ± 0	36 ± 9
0,2	0,5	67 ± 0	26 ± 9
0,2	0,7	33 ± 0	21 ± 9
0,2	0,9	30 ± 0	16 ± 1
0,3	0,1	100 ± 0	21 ± 9
0,3	0,3	93 ± 0	26 ± 18
0,3	0,5	67 ± 0	16 ± 1
0,3	0,7	41 ± 0	26 ± 9
0,3	0,9	30 ± 0	239 ± 374
0,4	0,1	100 ± 0	26 ± 9
0,4	0,3	89 ± 0	31 ± 1
0,4	0,5	70 ± 0	26 ± 9
0,4	0,7	37 ± 0	26 ± 9
0,4	0,9	30 ± 0	16 ± 1
0,5	0,1	100 ± 0	21 ± 9
0,5	0,3	89 ± 0	21 ± 9
0,5	0,5	70 ± 0	15 ± 1
0,5	0,7	52 ± 0	21 ± 9
0,5	0,9	30 ± 0	21 ± 9
0,6	0,1	100 ± 0	26 ± 9
0,6	0,3	96 ± 0	47 ± 41
0,6	0,5	74 ± 0	26 ± 9
0,6	0,7	48 ± 0	21 ± 9
0,6	0,9	33 ± 0	26 ± 10
0,7	0,1	100 ± 0	26 ± 9
0,7	0,3	96 ± 0	31 ± 1
0,7	0,5	74 ± 0	31 ± 1
0,7	0,7	48 ± 0	26 ± 9
0,7	0,9	33 ± 0	20 ± 9
0,8	0,1	100 ± 0	26 ± 9
0,8	0,3	96 ± 0	26 ± 9
0,8	0,5	74 ± 0	20 ± 9
0,8	0,7	48 ± 0	21 ± 9
0,8	0,9	37 ± 0	26 ± 9
0,9	0,1	100 ± 0	21 ± 9
0,9	0,3	96 ± 0	21 ± 10
0,9	0,5	74 ± 0	26 ± 10
0,9	0,7	56 ± 0	21 ± 9
0,9	0,9	30 ± 0	20 ± 9

SVM

In Knime, the SVM Kernel choosen can be Polynomial, Hyper tangent or RBF. The SVM method performance was evaluated in terms on training time and accuracy for dataset A3 and B3. The parameters configured for the available kernels were: Polynomial Kernel (Polynomial power tested from 1 to 10) and RBF Kernel (Sigma from 0.1 to 2.1 in steps of 0.2). The Hyper Tangent Kernel was also tested but it was not very accurate and therefore results are not presented in the following table.

Table 3.11 shows a summary of results, tests were executed three times and values presented are the average. The best higher values of accuracy, (100%), were obtained using a SVM that implements a polynomial Kernel with a polynomial degree higher than 4, and a RBF Kernel with a RBF Sigma value lower than 0.5. Among all these cases, the one with the lowest training time was RBF with a sigma of 0.1.

Table 3.11: Knime SVM performance evaluation (Knime).

Dataset	Kernel Type	Polynomial Power	RBF Sigma	Avg. Accuracy [%]	Avg. Training Time [ms]
A3	Polynomial	2	-	70 ± 0	130 ± 9
A3	Polynomial	3	-	96 ± 0	244 ± 59
A3	Polynomial	4	-	100 ± 0	198 ± 45
A3	Polynomial	5	-	100 ± 0	151 ± 9
A3	Polynomial	6	-	100 ± 0	114 ± 24
A3	RBF	-	0,1	100 ± 0	52 ± 18
A3	RBF	-	0,3	100 ± 0	57 ± 18
A3	RBF	-	0,5	100 ± 0	83 ± 9
A3	RBF	-	0,7	96 ± 0	109 ± 1
A3	RBF	-	0,9	96 ± 0	83 ± 24
A3	RBF	-	1,1	78 ± 0	62 ± 16
A3	RBF	-	1,3	70 ± 0	73 ± 9
B3	Polynomial	2	-	48 ± 0	68 ± 24
B3	Polynomial	3	-	85 ± 0	748 ± 12
B3	Polynomial	4	-	100 ± 0	1557 ± 41
B3	Polynomial	5	-	100 ± 0	3226 ± 246
B3	Polynomial	6	-	100 ± 0	5427 ± 230
B3	RBF	-	0,1	100 ± 0	77 ± 12
B3	RBF	-	0,3	100 ± 0	73 ± 9
B3	RBF	-	0,5	100 ± 0	92 ± 14
B3	RBF	-	0,7	93 ± 0	77 ± 8
B3	RBF	-	0,9	89 ± 0	89 ± 15
B3	RBF	-	1,1	70 ± 0	111 ± 10
B3	RBF	-	1,3	70 ± 0	104 ± 13

Fuzzy Logic

The Fuzzy Rule model creates a set of rules for incoming arrays based on some threshold values. The performance of different fuzzy normalizations and shrink functions is tested limiting the number of iterations to 50. Performance results are presented in Table 3.12, where it is observed that all methods perform very similar in terms of accuracy and training time for different parameters. Fuzzy Logic method is clearly affected by the presence of noise in the dataset, since the accuracy decreases from 100% when no noise is present (Dataset A3) to 41% when the dataset has a 30% of noise level (Dataset B3).

Table 3.12: Fuzzy Rule Learning performance (Knime).

Dataset	Shrink	Norm	Avg. Accuracy [%]	Avg. Training Time [ms]
A3	Volume Border Based	Min/Max	100 ± 0	23 ± 6
A3	Volume Anchor Based	Min/Max	100 ± 0	30 ± 0
A3	Volume Rule Based	Min/Max	100 ± 0	23 ± 6
A3	Volume Border Based	Product Norm	100 ± 0	20 ± 0
A3	Volume Anchor Based	Product Norm	100 ± 0	23 ± 6
A3	Volume Rule Based	Product Norm	100 ± 0	23 ± 6
A3	Volume Border Based	Lukasiewicz	100 ± 0	23 ± 6
A3	Volume Anchor Based	Lukasiewicz	100 ± 0	27 ± 6
A3	Volume Rule Based	Lukasiewicz	100 ± 0	23 ± 6
A3	Volume Border Based	Yager [0.2]	100 ± 0	30 ± 0
A3	Volume Anchor Based	Yager [0.2]	100 ± 0	30 ± 0
A3	Volume Rule Based	Yager [0.2]	100 ± 0	20 ± 0
B3	Volume Border Based	Min/Max	41 ± 0	37 ± 6
B3	Volume Anchor Based	Min/Max	41 ± 0	47 ± 21
B3	Volume Rule Based	Min/Max	41 ± 0	40 ± 20
B3	Volume Border Based	Product Norm	41 ± 0	37 ± 6
B3	Volume Anchor Based	Product Norm	41 ± 0	30 ± 0
B3	Volume Rule Based	Product Norm	41 ± 0	33 ± 6
B3	Volume Border Based	Lukasiewicz	41 ± 0	33 ± 6
B3	Volume Anchor Based	Lukasiewicz	41 ± 0	30 ± 10
B3	Volume Rule Based	Lukasiewicz	41 ± 0	45 ± 17
B3	Volume Border Based	Yager [0.2]	41 ± 0	41 ± 11
B3	Volume Anchor Based	Yager [0.2]	41 ± 0	40 ± 9
B3	Volume Rule Based	Yager [0.2]	41 ± 0	37 ± 6

Decision Tree (DT)

The performance of decision trees is evaluated in terms of accuracy and training time when changing the quality measure (Gain ratio or Gini Index) for a fixing pruning method of MDL. The depth of the tree is determined by the minimum number of records per leave, which is set to 2. Table 3.13 shows the performance results. Here is seen that the method training time is very short and the model gives very accurate results. In this case, the minimum records per node was set to a low value (2) which may have the impact of causing overfitting. The selection of this value must be careful chosen when using real datasets.

Table 3.13: Decision tree performance (Knime).

Dataset	Quality measure	Avg. Accuracy [%]	Avg. Training Time [ms]
A3	Gini	100 ± 0	26 ± 9
A3	Gain Ratio	100 ± 0	21 ± 9
B3	Gini	100 ± 0	26 ± 9
B3	Gain Ratio	100 ± 0	26 ± 9

Overall comparison (Knime)

Different machine learning methods were evaluated using the data mining software Knime: RProp MLP, PNN, SVM, Fuzzy rules and Decision Trees. These were trained using the datasets previosuly presented in Table 3.2 and were evaluated in terms of training time, accuracy and final model size. For each model the optimal configuration of internal parameters that achieve the higher accuracy and lower training time was found. The configuration parameters, and the number of training iterations for each method is presented below:

- RProp MLP: Maximum number of training iterations=200, maximum number of hidden layers = 10 (Increment +1 in each iteration), maximum number of hidden nodes per hidden layer=40 (increment +2 per iteration)
- PNN: Maximum number of training iterations=50, theta minus=0.1 and theta plus=0.6.
- SVM: RBF Kernel with sigma = 0.1.
- Fuzzy Rule: Maximum number of training iterations of 100, fuzzy normalization: Product, shrink function: Volume Border Based.
- Decision Tree: Gini quality measure and MDL prunning.

Figure 3.9 shows the accuracies of the different machine learning methods. It is seen that for datasets type A most of the methods achieve 100% accuracy. When dataset type B is used, some methods as MLP and Fuzzy are less accurate. From these results, SVM, PNN and DT are the best performing methods.

Figure 3.9: Accuracy obtained by different methods (Knime).

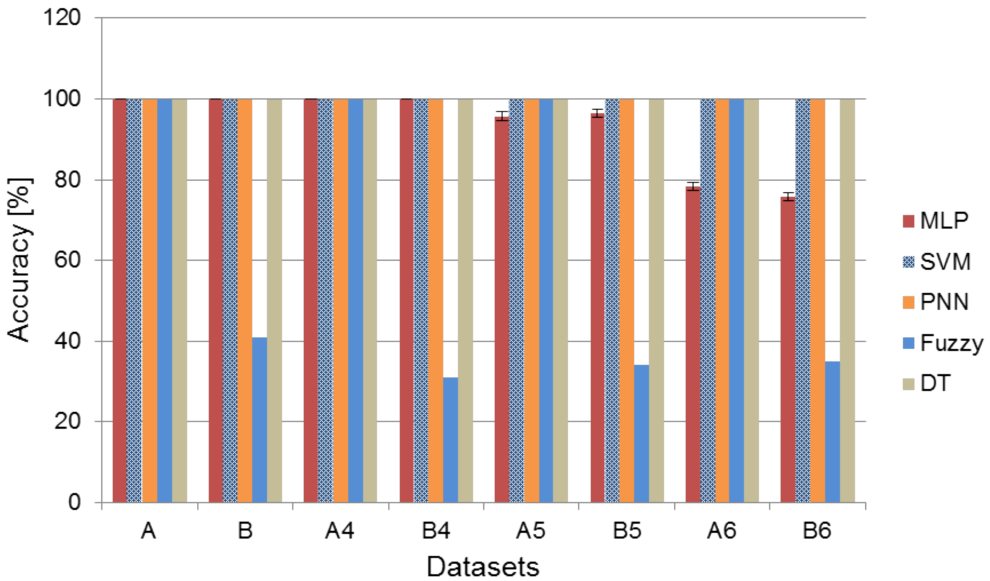
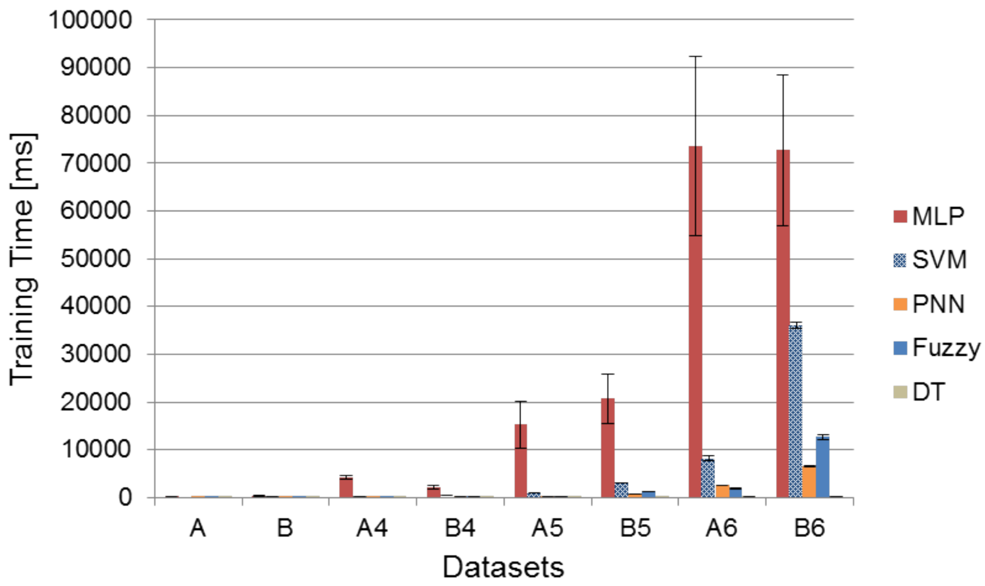


Figure 3.10 shows a comparison of the different machine learning methods in terms of training time. A similar pattern to the obtained with Encog is observed (Figure 3.5); here the training time fits quite an exponential function with respect to the dataset size and number of inputs. From this graph is observed that the methods with lower training times are PNN and DT.

Figure 3.10: Training time consumed by different methods (Knime).



3.4 Summary

This section presents an overall introduction to the proposed system. Three architectures oriented to ease the connection between different devices are explained. A mixed architecture is selected as the appropriated thanks to the advantages it provides in terms of distribution of processing load, redundancy and future scalability simplicity. The section also presents available communication mechanisms used for accessing vehicle information and data coming from other devices.

The section also explore the benefits of different data mining software (Encog and Knime) when it respect to data analysis and training of machine learning methods. Different machine learning methods are compared in terms of training time and prediction accuracy. As conclusion after the tests, the software Knime is selected as preferred tool due to the great amount of functionalities available and a easier block programming oriented design. In regards of machine learning performance, PNN and Decision trees have shown to be the more adequate for the tested dataset.

Chapter 4

Estimating Driving Complexity

This section presents the procedure undertaken for creating model capable of estimating the driving complexity in real time. The proposed system was designed to minimize the investment needed and ease the integration in current vehicles; no additional hardware is required to be introduced into vehicles, since input variables are already accessible and the required computing power for calculating the complexity is significantly low.

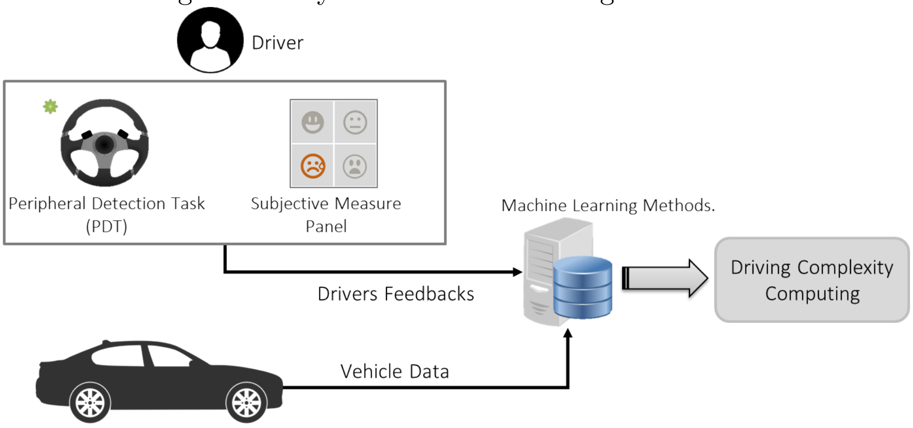
The model was created taken as baseline previous researches (see Section 2.2.4). In these studies, the estimation was computed as a function of several variables considered as informative of the driving complexity scenario. In this regards, the methodology employed consisted in monitoring a set of variables in real driving scenarios and identifying existing patterns in terms of two metrics: User Subjective Feedback and Response Time.

User Subjective values are given by the own users' perception about the driving complexity, while response time is measured as the reaction time between the trigger of a sporadic event and the proper driver execution of a predefined task. From the subjective field, a panel in which the driver could select its current perceived subjective workload based in the scenario was incorporated. With regards to measuring the Response Time, a Peripheral Detection Task (PDT) was used as indicator. Finally, at the end of each test, a questionnaire was presented to users for having a general understanding of the complete trip.

Among the different variables that can be used as a reference for estimating the complexity of driving, the most informative variables were selected to be monitored in the final model. Driving performance and environmental metrics seem to be more correlated to driver workload since these define and characterize the driving scenario. Physiological variables were not considered, in part due to their intrusively and in part due to its sensibility to emotional strains non related to driving workload.

An scheme of the proposed system used for creating the model is summarized in Figure 4.1. Variables obtained from the vehicle (Vehicle Data) and Drivers Feedbacks were used to create a dataset of different driving scenarios. Once data was collected, data mining techniques were employed for constructing the estimator model of driving complexity. Each variable was treated independently; this approach let the machine learning algorithm be the one in charge of assigning weights and priorities to each variable.

Figure 4.1: System used for creating the model.



4.1 Methodology

As previously stated, many variables have an impact on driving complexity and the degree of influence of each variable is hard to estimate. Therefore, in order to create a model capable of making estimations, the approach followed consisted in the use of data mining techniques and machine learning methods that can be trained with multiple datasets. The advantage of using a model obtained from machine learning methods as a driving complexity estimator is that it can easily be updated when incorporating new variables, as well as being capable of improvement when more data is available.

The data required for training the machine learning methods was collected in real driving scenarios. Several participants were requested to drive an instrumented vehicle in different scenarios and to perform two simple tasks while driving: (1) report their driving complexity perception every time they felt a change using a panel installed in the vehicle, and (2) press a button located in the steering wheel each time they detected that LEDs installed in the windshield were turned on at random intervals (PDT). If participants did not report the perceived complexity of driving for more than 5 minutes, the panel emitted a short beep as a reminder.

The purpose of the PDT mechanism was to measure variations in the response time due to different complexities in the driving scenario. As previously explained, the PDT task goes trough the same channels as some of the driving tasks, therefore increments in the driving complexity are expected to cause interference in the adequate performing of the PDT task. PDT buttons were located next to steering wheel buttons to avoid safety implications during the test.

The system installed in the vehicle consisted of a Linux machine directly connected to the CAN bus in order to access vehicle variables, and to control the subjective panel and the PDT. Participants were asked to concentrate on the driving and execution of the test and to avoid any use of the infotainment system that could affect road safety.

The selection of routes was defined with each participant in order to obtain diverse driving scenarios with different road types. Most of the routes consisted of their habitual journeys between home and work and vice-versa (personal routines), while other routes were already familiar to them but not habitually used. From the total dataset, the percentage of usual routes rises up to 80% of the total, and the percentage of known routes to 96%, which can be considered as homogeneous enough for the purpose of the test. Based on (Krumm

(2012)), most of the trips undertaken by people are known, and the percentage of what we regard as the most frequently used routes account for approximately 70% of the total.

During the trip, driving parameters and selected environment variables were recorded at a sample rate of 3Hz. Once data were collected, data mining techniques were used for identifying the variables that had a higher impact on driving complexity based on the feedbacks used (user subjective perception and response time to the PDT). These relevant variables were used as inputs for creating the model implementing machine learning methods. For analysing data and creating the models, the data mining software Knime v3.1.1 was employed.

Since not all machine learning methods behave in the same manner, different methods were tested in order to select the most suitable one for this purpose in terms of estimation accuracy and training time. The resulting model can be used as the basis for categorizing the driving complexity scenario. The machine learning methods compared are the following: Decision Trees, Probabilistic Neural Networks (PNN) and Random Forests.

4.1.1 Participants

As this study do not seek a profile categorization, participants were selected on the basis of achieving a homogeneous profile in terms of driving experience and driving periodicity. They were all regular drivers who had held a driving license for at least three years and had driven more than 30.000 Kms during the course of their lives. The age of the participants ranged from 22 to 40. Table 4.1 presents profile categories arranged according to gender, age and driving style. The driving style field is defined according to the subjective appreciation of the drivers themselves.

Table 4.1: Participants' Profile.

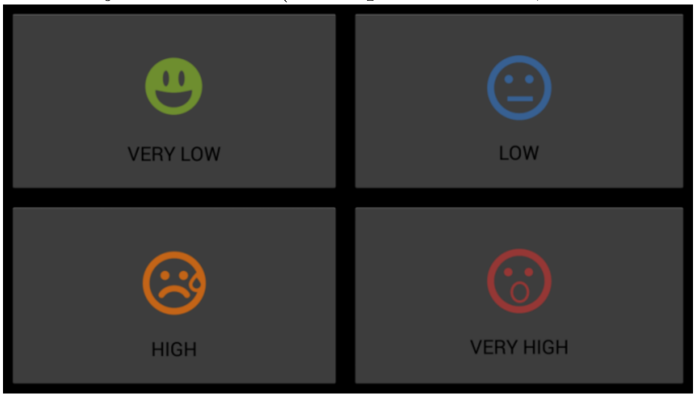
Gender	Female(8), Male (12)
Age	Average: 29, SD:4
Driving Trips per month	Average: 60
Driving Style	Normal (14), Slow(2), Sportive (4)

Participants were initially introduced to the test and were requested to drive for 30 minutes while under supervision in order to ensure that they understood the test procedure. This time also served as a training phase for participants to practice their reaction to PDT (these data were not taken into account in the statistics). Participants' behaviour was not recorded by any camera during the test, data was gathered only from the vehicle and driver feedback reports (response time and subjective perception). Participants were asked to follow the required safety regulations at all times, to avoid using the infotainment system of the vehicle and to concentrate on the road scenario and the execution of the test.

4.1.2 System

The system installed in the vehicle consisted of a Linux machine connected directly to the CAN bus in order to access the complete set of variables available in a vehicle. The machine was also connected to a micro-controller in charge of controlling the PDT interface as well as to a mobile phone used as Subjective Panel. The panel used for reporting state changes is presented in Figure 4.2, here 4 boxes are used to identify each one of the subjective states perceived driving complexity (Very Low, Low, High and Very High). The state indicated by the participant was kept highlighted to remember the user the actual state.

Figure 4.2: Subjective Panel (Smartphone Screen, all states disabled).



A real implementation of the system from the participant perspective can be seen in Figure 4.3, where one may observe that the mobile phone works as panel for indicating complexity changes (4 levels), the two LEDs for presenting the PDT tasks, and the two buttons in the steering wheel for responding to these events. All elements were located in accessible places to ease the reporting of feedbacks and avoid an interference with primary driving tasks.

Figure 4.3: Testing System.

The set of monitored driving variables are shown in Table 4.2. All inputs are obtained from the vehicle except the Road Type variable, which is obtained from Open Street Map

(OSM)[1] giving the GPS coordinates of the vehicle navigation system. Data were classified according to driving seasons or trips where each trip consisted in a journey from the point of departure to the destination. The initial and final sections were removed in case they were associated to parking manoeuvres, since the model was not designed for calculating complexities in these types of driving scenarios.

Table 4.2: Set of variables used.

Type	Name	Explanation	Unit	Range
Input	Travel_Time	Current travel time	seconds	-
	Time_of_Day	Time of day expressed in minutes	minutes	0 to 1440
	Speed	Travel speed	Km/h	0 to 320
	WheelAngle	Steering wheel rotation angle	degrees	0 to 800
	Brake_pressure	Brake pedal pressure	Bar	-30 to 276
	Acceleration_Pressure	Acceleration pedal pressure	%	0 to 100
	Temperature	Environment temperature	°C	-50 to 76
	Road_SpeedLimit	Speed limit of road	Km/h	-
	Lat_acceleration	Vehicle lateral acceleration	G Force	-1.27 to 1.27
	Long_acceleration	Vehicle longitudinal acceleration	m/s2	-16 to 15.9
	DTL_m	Proximity to border line	meters	-1.25 to 1.25
	ACC_Abstandsindex	Distance to front vehicle in a categorical index	index	1 - 16
	Ambient_Light	Environmental light in [Lux] (range between 100-38000 or outside)	index	0 or 1
	DayNight	Day or Night defined by time of day	index	0:Day, 1:Night
	RainIntensity	Rain intensity (vehicle sensor and wipers position)	index	0:Low, 1:Medium, 2:High
	NavLatitude	Coordinate latitude GPS	°	-
	NavLongitude	Coordinate longitude GPS	°	-
	Road_Type	Road definition given by OSM database	OSM Type	0:Unknown 1:Motorway 2:Motorway Link 3:Primary 4:Primary Link 5:Secondary 6:Secondary Link 7:Tertiary 8:Tertiary Link 9:Residential
	SpeedLimit	Speed limit indicated by road signs	Km/h	-
	participantId	Participant identification	index	0 to 20
	TripId	Trip identification	index	0 to 126
Feedback	User_Subjective	Perceived driving complexity by participant	Index	0:Very Low 1:Low 2:High 3:Very High
	Response_Time	Response time on PDT events	msecs	0 to 5000

A temporal characterization of the scenario for the previous seconds was also included as input variables. These additional variables are the mean, standard deviation, range, maximum value and minimum value for the last 3, 7 and 15 seconds. The purpose of adding this variables was to achieve a more stable representation of the scenario, since specific events may not always be reported or perceived as a change in current driving complexity; e.g., a curve scenario composed by a sequence of bends is expected to be reported as more complex scenario that one containing a single bend in a road.

Roads were classified on the basis of road link type (e.g., Primary or Primary Link), since this may help to improve the categorization of the road topology and analyse the influence of incorporations and exits of roadways. Some statistical crash reports (European Commission (2015a)) detected a different number of crashes or accidents depending on the presence and type of junctions.

Physiological feedbacks were not monitored, due in part to their intrusiveness and also because of sensitivity to emotional strains that are not related to the driving workload. Regarding environmental variables, we assume that weather conditions also affect the driving

[1]More information in this regards is added in the Appendix Section (D).

complexity. Nevertheless, even when these type of variable were monitored by the system, it was not incorporated into the final model due to small variations recorded in weather conditions during the execution of tests.

4.2 Results

The dataset was constructed gathering 103 hours of driving over a distance of 6,460 Kms at a sample time of 3 Hz. This section initially presents an overall statistical analysis of data collected. Secondly, data is pre-processed in order to remove unnecessary parameters or variables and achieve a dimensionality reduction. According to (Values et al. (2014)), dimensionality reduction is not only useful for speeding up the training of the model, but also help in the classification accuracy. Noisy or faulty input data often lead to a decay of the algorithm performance. The study cited also presents a list of techniques recommended to achieve a good dimensionality reduction.

Once the dataset has been filtered, different machine learning methods are trained and compared in terms of accuracy and training time. Finally, the best performing model is validated using as dataset the driving perception of an expert in safety when driving in different scenarios.

4.2.1 Data overview

Table 4.3 shows statistical information for each variable recorded during the trips. A wide range of values for each variable is observed (range as values between Min and Max values), which increases the probability of having more diverse scenarios. Diverse datasets are advantageous when creating the model, since the larger the amount of scenarios known, the lower will the uncertainty and the number of unknown values the model has to estimate.

The table also presents information in regards of the skewness factor, which indicates the symmetry of values distribution, and the Kurtois factor, which measures the tail deviation in relation to a normal distribution. Among all the variables presented, those associated to rain, wheel angle and brake pressure yield higher values. These results are attributed to fewer rainy conditions, common straight roads and infrequent strong pressure while braking; which is expected in most driving scenarios.

Driving Complexity was analysed in terms of two feedback variables: User Subjective and Response Time. These feedbacks are not expected to be globally correlated to unique driving complexities, since each participant has different perceptions of risk and physical conditions, which give rise to variations in response times even for the same driving complexities. Instead, the main goal is to identify overall trends that can be used for defining an overall predictive model that suits to most drivers.

Table 4.4 provides a summary of the subjective workload perceived by each participant during the trips; it was observed that participants tend to perceive low workload states given by Very Low and Low indications (representing 77.55% of the total). This result may be associated to driving confidence or suitable driving conditions relating to the volume of traffic, road type and good weather conditions. All participants perceived every possible driving complexity (from very low to very high), although participants 3, 4, 7, 16 and 17 evidently encountered few cases of very high complexity.

In a Similar way, the Response Time feedback for each participant is shown in Table 4.5. Since the PDT task was randomly activated, only the data around the moment the event was triggered was taken into account in the analysis, more specifically, the 5 seconds

Table 4.3: Statistics of variables.

Variables	Min	Max	Mean	SD	Skewness	Kurtosis
Travel_Duration	453	10220	1880.9	1790.3	1.67	2.79
Speed	0	179	72.67	41.83	-0.16	-1.16
WheelAngle	0	496	13.29	40.59	7.31	65.12
Brake_pressure	-1	109	1.51	5.37	5	32.35
Acceleration_pressure	0	100	22.43	22.15	0.51	-0.68
Lat_acceleration	0	0.9	0.04	0.08	2.35	7.02
Long_acceleration	0	8.7	0.38	0.47	2.48	9.33
DTL_m	-1.3	1.3	-0.68	0.44	0.21	0.08
ACC_Abstandsindex	1	16	10.92	6.05	-0.64	-1.27
Ambient_Light	0	100	43.48	49.57	0.26	-1.93
RainIntensity	0	2	0.02	0.18	8.44	76.92
NavLatitude	39.16	42.41	41.45	0.43	-2.29	9.42
NavLongitude	-0.55	2.83	1.82	0.53	-2.09	5.08
Temperature	-2	23	11.21	4.65	-0.05	-0.49
Response_Time	0	5000	1950.58	1375.77	1.33	0.46
User_Subjective	1	4	2.06	0.84	0.44	-0.39
Time_of_Day	390	1380	826.72	270.42	0.02	-1.41
SpeedLimit	0	120	66.53	48.25	-0.27	-1.48
RoadType	0	9	3.16	2.67	0.89	-0.53
Day/Night	0	1	0.19	0.4	1.55	0.4

Units given in 4.2.

Table 4.4: User Subjective by participant [%].

Participant Id	Very Low	Low	High	Very High
1	22.53	47.7	21.85	7.92
2	4.25	39.02	49.86	6.87
3	43.06	53.13	3.71	0.1
4	24.84	67.79	7.33	0.04
5	7.07	45.83	38.96	8.14
6	19.02	50.62	21.77	8.59
7	62.49	29.52	7.52	0.47
8	43.82	46.08	8.49	1.61
9	13.36	39.54	40.34	6.77
10	25.43	39.33	27.23	8.01
11	21.99	56.47	17.16	4.37
12	20.9	47.35	21.22	10.53
13	33.84	52.78	11.75	1.63
14	13.15	50.27	34.6	1.98
15	19.27	47.56	29.35	3.83
16	12.02	76.59	11.38	0.01
17	46.33	41.2	11.95	0.52
18	23.33	48.24	22.87	5.56
19	22.61	25.23	31.87	20.29
20	46.31	29.48	18.05	6.15
Total	26.55	46	21.95	5.5

Table 4.5: Response Time in msecs by participant [%].

Participant ID	<1000	1000-2000	2000-3000	3000-4000	4000-5000	>5000 (Timeout)
1	36.14	36.06	10.95	3.86	2.8	10.19
2	8.48	69.28	13.39	1.64	2.06	5.15
3	12.4	59.48	16.73	4.73	1.2	5.46
4	5.14	76.39	9.96	2.47	1.24	4.8
5	56.94	32.89	4.88	1.95	1.55	1.78
6	11.1	47.84	19.44	6.78	2.82	12.03
7	12.9	76.28	6.07	0.92	0.92	2.9
8	19.35	57.95	11.08	1.29	1.94	8.39
9	3.76	53.99	17.67	4.69	1.88	18.01
10	4.78	62.93	12.44	4.84	3.68	11.33
11	50.18	38.09	5.95	1.44	0.51	3.82
12	3.07	63.58	12.0	6.84	2.25	12.26
13	52.73	35.93	5.11	0.73	2.19	3.31
14	1.28	25.02	20.17	16.68	9.18	27.67
15	23.36	44.73	14.6	4.24	3.18	9.89
16	58.28	27.99	5.19	2.06	1.3	5.19
17	20.53	70.68	5.73	0.66	0.0	2.4
18	7.58	55.17	13.53	4.34	2.78	16.61
19	20.03	57.28	8.34	3.58	1.66	9.11
20	33.93	38.3	8.02	6.54	1.48	11.73
Total	20.92	53.12	11.09	3.88	2.11	8.89

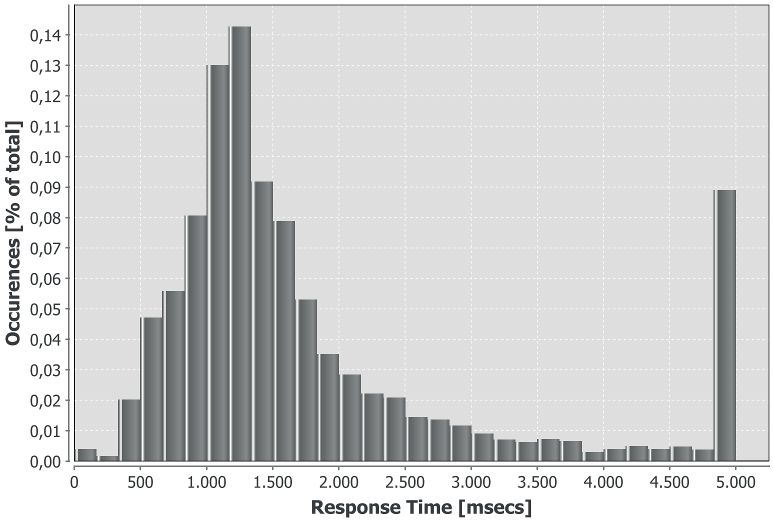
before and the second after the participants detected the event. In general, participants tend to detect events before 2,000 msecs (74.04% of total). It is also observed that the number of delayed responses decreases from 2,000 msecs to 5,000 msecs and then rises at the end (responses times higher than 5000ms were grouped and considered timeouts). From the participants' answers when asked about this aspect, it was found that most of these timeouts were associated with failure to pay attention to the PDT rather than the inability to press the button due to complex scenarios.

In general, participants tend to have similar responses times with a low number of timeouts and a high probability of responding before 3 seconds. Some exceptions are found for participant 14, who gave a greater number of delayed responses; this may be the consequence of a higher complexity in the scenario or frequent distractions and failure to respond to the PDT.

In order to gain a more graphic understanding of the distribution of response times, a histogram of the recorded values is shown in Figure 4.4. Here one may observe that the Response Time of participants tends to a skewed distribution (ignoring last bar) with a higher density of values between 1,100 and 1,500 milliseconds. Based on previous researches, this indicates that participants were easily able to identify the majority of events (a fast detection is expected when responses values are inferior to 1.5 seconds, (Dewar and Olson (2002)). For this dataset, the amount of response times lower than 1500 milliseconds accounts up to 57% of total.

Observation of the distribution suggests that it is convenient to remove timeout events from the analysis, since otherwise it is not possible to distinguish real driving distraction from lack of attention to the PTD by participants. It is for this reason that the following datasets referring to the Response Time do not include these timeout values associated to response times higher to 5000ms.

Figure 4.4: Histogram of Response Time.



Comparison of User Subjective and Response Time variables yields a slightly positive linear correlation, which is expected because an increase in response time should be associated to an increase in the perceived complexity by the driver. A linear regression function after normalization is given in Equation 4.1 where the fitness R^2 is equal to 0.0046. The small coefficient may be related to two factors; the different perceptions of risk by participants and their different physical skills for responding to randomly activated events.

$$User\_Subjective = 0.0678 * Response\_Time \tag{4.1}$$

With regard to road diversity, Table 4.6 presents the percentage of road types driven by each participant. Route Type classifications were obtained from Open Street Maps (OSM) by taking the GPS coordinates recorded during the trips. Most driving scenarios were associated to motorways (45.46%) or secondary roads (15.73%), and thus the final model will provide a better understanding of these roads rather than others with a lower presence.

In the case of primary and tertiary roads, the percentage of data gathered is 10.53% and 10.57%, respectively. However, some participants (3, 4, 11, 17 and 20) did not drive often on these roads, which means that the model may to some extent be biased towards other participants who did drive in these routes.

Table 4.6: Road Types per participant [%].

Participant Id	Unknown (0)	Motorway (1)	Motorway links (2)	Primary (3)	Primary link (4)	Secondary (5)	Secondary link (6)	Tertiary (7)	Tertiary link (8)	Residential (9)
1	0.18	31.2	3.39	8.63	0.41	9.41	0.0	41.4	0.46	4.91
2	1.69	48.29	13.42	3.55	0.52	20.85	0.0	4.47	0.0	7.22
3	0.97	82.53	5.79	3.89	0.01	2.34	0.0	0.13	0.03	4.3
4	0.2	89.42	1.94	0.0	0.0	5.16	0.0	1.83	0.0	1.45
5	0.84	35.86	8.64	17.64	0.03	17.67	0.0	8.36	0.59	10.37
7	1.59	42.54	6.54	1.36	0.04	15.74	0.0	17.82	0.0	14.37
9	0.8	44.85	8.31	14.49	1.15	11.07	0.24	12.0	0.06	7.02
11	0.89	12.18	12.1	2.38	0.34	59.94	0.0	2.8	0.0	9.38
13	1.46	35.43	10.04	5.08	0.28	17.48	0.38	21.39	0.09	8.35
14	4.2	15.72	3.95	4.3	0.0	18.67	0.12	29.86	0.01	23.16
15	7.79	36.34	5.0	10.39	0.4	13.34	0.0	17.32	0.0	9.42
16	2.09	43.19	16.9	13.89	0.83	7.93	0.0	7.26	0.26	7.66
17	0.3	72.36	12.29	0.71	0.0	9.78	0.0	2.13	0.01	2.42
18	1.83	56.77	7.93	4.77	0.0	8.75	0.08	9.11	0.13	10.63
19	0.92	50.63	12.95	3.24	0.07	22.14	0.0	7.35	0.17	2.53
20	1.75	20.71	1.09	62.65	0.45	7.69	0.0	2.2	0.0	3.45
Total	1.71	45.46	7.91	10.53	0.3	15.73	0.04	10.57	0.28	7.47

It is observed that participants 3, 4 and 17 have driven mostly on motorways. While referring back to Table 4.4 shows that these participants belonged to the group that did not perceive very complex scenarios (3,4,7,16 and 17). It is therefore expected that motorways scenarios probably have a lower perceived complexity than other types of roads.

A similar relation can be seen in the Response Time, where participant 14, who had higher Responses Time (see Table 4.5), is also the one who drove along more residential and tertiary roads. This global analysis shows that road type is expected to play an important role in the categorization of driving complexity.

4.2.2 Data pre-processing

Pre-Processing of data consists in removing parking manoeuvres, standardizing variables (Z-Normalization), removing high correlated variables, balancing the dataset in relation to classes and performing dimensional reduction techniques based on the recommendations (Values et al. (2014)). Parking manoeuvres were removed since the model was not designed to predict this kind of scenarios, variables were normalized in order to set different the variables to a common scale, balancing is performed since some machine-learning methods are biased towards the most frequent class, and the dimensionality reduction helps accelerating training time and improving classification.

The analysis is performed according to User Subjective feedback and Response Time feedback. User Subjective feedback is defined in terms of four nominal values, and therefore will be studied using classification methods based on four main classes: 1 for Very Low, 2 for Low, 3 for High, and 4 for Very high. On the other hand, Response Time feedback is a continuous value and its analysis is performed using regression methods. All the machine learning methods evaluated are capable of dealing with both training schemes.

The balancing of data can be performed by sub-sampling majority classes, in which some valuable information may be lost, or by over-sampling minority classes, where erroneous categorizations in the minority class may be propagated. In the dataset presented here, the minority class corresponding to a "Very High" driving complexity represents just 5.5% of the data. In order to increase the minority class presence, and to prevent the propagation of any possible erroneous categorizations, the balancing procedure involves the duplication of this minority class once and then then sub-sample majority classes by removing rows of each one in a random manner.

As a result of this procedure, the dataset was reduced to 38.8% of it initial size, from 1,114,052 rows (around 103 hours) to 432,496 rows (around 40 hours)[2]. This reduction in the dataset size does not entails a significant lost of information since the meaning values of input variables do not have a great variability. A 3Hz sampling rate adds enough redundancy information to allow the removing of certain rows without affecting the knowledge present in the data. Regarding Response Time, the dataset was not balanced because this feedback is continuous in time and regression methods are used.

The application of dimensional reduction techniques was mandatory since the dataset contained more than 160 variables, some of which were highly correlated, while others may exert no influence on the categorization of the driving scenario. The purpose of this process was to simplify the model, reduce the algorithm execution time and to avoid a possible over-fitting. The techniques applied for this purpose were: analysis of the linear correlation between variables and feedback; the generation of a tree ensemble for finding the most informative set of features, and finally the execution of a feature-back elimination process using Decision Trees.

Linear correlation

Linear correlations associated to User Subjective and Response Time variables for the entire dataset are given in Table 4.7; only the 20 with highest correlation values are shown because of the large amount of variables measured. A linear correlation analysis is not expected to give enough information for creating the model, since variables are analysed independently and the feedback is expected to be given as a function of several variables. Nevertheless, it is advisable first to check which variables may have a greater influence. In this case, one of the higher correlated variables is associated to speed.

For both feedbacks used, User Subjective and Response Time, The values with higher correlations common for both are: Speed (Min 15secs), Speed (Min 7secs),Speed (Mean 7secs) which shows the relevance of the speed variable when it respect to assessing driving complexity. It also detected how variables with temporal memory (minimum, mean or maximum values in the last seconds) offer more information than punctual ones.

[2]Not continuous time representation since rows were selected randomly.

Table 4.7: Linear correlation of User Subjective and Response Time associated to inputs.

User Subjective		Response Time	
Variable	Correlation	Variable	Correlation
Speed (Min 15secs)	-0,304	ACC_Abstandsindex (Sum 15secs)	0,128
Speed (Min 7secs)	-0,303	ACC_Abstandsindex (Sum 7secs)	0,123
Speed (Mean 15secs)	-0,301	Speed (Mean 7secs)	-0,119
Speed (Min 3secs)	-0,301	Speed (Min 7secs)	-0,119
Speed (Mean 7secs)	-0,300	Speed (Max 3secs)	-0,119
Speed (Mean 3secs)	-0,299	Speed (Mean 3secs)	-0,119
Speed	-0,298	Speed (Max 7secs)	-0,118
Speed (Max 3secs)	-0,297	Speed (Mean 15secs)	-0,118
Speed (Max 7secs)	-0,296	Speed (Min 15secs)	-0,118
Speed (Max 15secs)	0,216	Speed (Min 3secs)	-0,118
Long_acceleration (Range 15secs)	0,214	Speed	-0,117
Long_acceleration (Max 15secs)	0,204	ACC_Abstandsindex (Max 7secs)	0,116
Long_acceleration (SD 15secs)	0,197	ACC_Abstandsindex (Max 15secs)	0,116
Long_acceleration (Mean 15secs)	0,190	Speed (Max 15secs)	-0,115
Long_acceleration (Max 7secs)	0,189	ACC_Abstandsindex (Sum 3secs)	0,114
Long_acceleration (Range 7secs)	0,182	ACC_Abstandsindex (Max 3secs)	0,113
Long_acceleration (SD 7secs)	-0,179	ACC_Abstandsindex (Mean 7secs)	0,112
SpeedLimit	0,170	ACC_Abstandsindex (Mean 15secs)	0,112
Long_acceleration (Mean 7secs)	0,164	ACC_Abstandsindex (Mean 3secs)	0,110
Long_acceleration (Max* 3secs)	0,155	ACC_Abstandsindex	0,109

As regards User Subjective feedback, a high negative correlation is observed for speed, which indicates that users tend to perceive lower driving complexity at higher speeds. After asking the participants some questions, it was found that this phenomenon is related to motorway scenarios with a low volume of traffic. On the other hand, lower speeds were usually associated to higher volumes of traffic or scenarios that required greater attention.

Similarly, the longitudinal acceleration is positive correlated to User Subjective; this variable reflects changes in speed and is related to traffic variability or the presence of bends in the road, which is also expected to be an indicator of driving complexity. As previously expressed, this correlation analysis just give some clues about possible relations and correlation values are not high enough to make general statements.

Where Response Time feedback is concerned, a higher correlation in terms of speed and the traffic indicator –given by the distance between vehicles travelling in front– is observed. As regards the speed correlation values, a negative correlation is also observed, as occurs with the User Subjective feedback; this effect is an indicator of how some variables have a similar influence on the feedback variables evaluated. In this case, a higher Response Time was recorded at lower speeds which is probably caused to the presence of more dynamic scenarios with curves or in urban areas.

Tree Ensemble

Tree ensembles are useful for finding the most informative features contained in a dataset. The training process consists in creating several models of Decision Trees using different features in each one (selection of candidates[3]). Once the models are created, the most used feature for executing first splits in data are identified. A score is then assigned to each feature in terms of the times it was selected for performing a split, divided by the number of times it was selected as candidate. Only the first three splits are taken into account in

[3]Candidate: feature that has been taken into account for constructing a model.

the scoring function (4.2). Also, the third-level splits are sub-weighted since they are less relevant for classifying data.

$$Score = \frac{splits(level0)}{candidates(level0)} + \frac{splits(level1)}{candidates(level1)} + 0.5\frac{splits(level2)}{candidates(level2)} \quad (4.2)$$

For modelling the User Subjective feedback, Decision Trees were based on a nominal classification. The configuration implemented a Gini index split criteria, which was set to a maximum depth of 3 levels, and 200 models were constructed randomly selecting 5 features in each one. In regards to the Response Time feedback, Decision Trees based on a regression scheme were built, mid split points were used, and the maximum tree depth was set to 3 levels. In this case, 2,000 models were constructed randomly choosing 5 features per tree.

This process was repeated 100 times (iterations) in order to obtain stable results in terms of the maximum, minimum, mean and standard deviation of scores computed. Table 4.8 shows the 40 features with the highest scores, where variables with higher scores are expected to have a greater influence on the categorization of the driving complexity. Several of these variables are associated to a same feature (base feature); e.g., speed is a base variable and its mean, max, min, and standard deviation are associated variables, and therefore are highly correlated. In the final model, it is simply necessary to include one or two variables associated to the same base feature.

Observing Table 4.8, different speed variables obtained the highest scores for both feedbacks[4]. This is an indicator of the relevance of speed when categorizing driving complexity. Other base variables that achieved high scores for both feedbacks were associated to road type, wheel angle and the traffic indicator (ACC_Abstandsindex).

The score obtained in this procedure does not necessarily ensure that the most useful features for categorizing driving complexity can be identified, since constructed trees are evaluated up to a split-depth of three, and other significant relations may exist at deeper levels. The main purpose of this technique is to gain an overall understanding of which features have the greatest impact on driving complexity and thus serve as a first-dimensional reduction technique.

Backward Feature Elimination

Once the results of the Tree Ensembles have been evaluated, a Backward Feature Elimination procedure is executed, which consists in the evaluation of Decision Tree models when each feature is left out just once. This method enables less relevant variables to be identified, since the accuracy of the Decision Tree when a low influential variable is removed should not vary significantly. The applied method consists in determining Decision Trees accuracy when one feature is removed each time (iteration), then removing the less informative feature pertaining to accuracy, and finally repeating the process until all features are removed.

This process is executed using as features the highest scores obtained from the tree ensemble technique. The Decision Tree used as the model for User Subjective feedback implements a Gini index quality measure, limits the depth of the tree to 12, and uses the mid-split point. The Decision Tree model used in the case of Response Time is configured using a regression scheme and limiting the number of splits to 11.

[4]The maximum score possible to get is 2.5, supposing the same feature is used in the first 3 splits.

Table 4.8: Feature Scores (Tree Ensemble).

User Subjective				Response Time			
Variable	Maximum	Mean	SD	Variable	Maximum	Mean	SD
Speed (Min* 15secs)	2.087	2.030	0.022	Speed (Max* 7secs)	1.368	1.368	0.000
Speed (Mean 15secs)	1.996	1.894	0.045	Speed (Mean 3secs)	1.314	1.314	0.000
Speed (Min* 7secs)	1.968	1.860	0.058	Speed (Max* 3secs)	1.311	1.311	0.000
Speed (Mean 7secs)	1.967	1.815	0.061	Speed (Mean 7secs)	1.298	1.298	0.000
Speed (Min* 3secs)	1.867	1.696	0.079	Speed (Mean 15secs)	1.246	1.246	0.000
Speed (Max* 15secs)	1.816	1.617	0.076	Speed	1.243	1.243	0.000
Speed (Mean 3secs)	1.802	1.610	0.080	RoadType	1.233	1.233	0.000
Speed (Max* 7secs)	1.754	1.561	0.090	Speed (Min* 15secs)	1.233	1.233	0.000
Speed (Max* 3secs)	1.742	1.520	0.077	ACC_Abstandsindex (Sum 15secs)	1.229	1.229	0.000
Speed	1.734	1.486	0.086	Speed (Min* 7secs)	1.227	1.227	0.000
WheelAngle (Max* 15secs)	1.274	1.131	0.056	WheelAngle (Range 7secs)	1.209	1.209	0.000
WheelAngle (Range 15secs)	1.272	1.091	0.064	WheelAngle (Max* 7secs)	1.180	1.180	0.000
WheelAngle (Mean 15secs)	1.159	1.041	0.050	ACC_Abstandsindex (Sum 7secs)	1.179	1.179	0.000
WheelAngle (Range 7secs)	1.168	1.021	0.065	Speed (Min* 3secs)	1.163	1.163	0.000
WheelAngle (Max* 7secs)	1.143	0.995	0.054	ACC_Abstandsindex (Max* 15secs)	1.138	1.138	0.000
Long_acceleration (Range 15secs)	1.145	0.990	0.061	WheelAngle (Max* 15secs)	1.123	1.123	0.000
WheelAngle (SD 15secs)	1.117	0.975	0.061	Speed (Max* 15secs)	1.106	1.106	0.000
Long_acceleration (Max* 15secs)	1.177	0.953	0.056	WheelAngle (Range 15secs)	1.086	1.086	0.000
Lat_acceleration (Range 15secs)	1.049	0.933	0.048	ACC_Abstandsindex (Sum 3secs)	1.058	1.058	0.000
SpeedLimit	1.018	0.920	0.044	WheelAngle (Max* 3secs)	1.054	1.054	0.000
WheelAngle (SD 7secs)	1.107	0.918	0.061	DTL_m (Mean 15secs)	1.028	1.028	0.000
Lat_acceleration (Range 7secs)	1.038	0.889	0.060	ACC_Abstandsindex (Max* 7secs)	0.994	0.994	0.000
RoadType	1.026	0.879	0.060	WheelAngle (SD 7secs)	0.994	0.994	0.000
Long_acceleration (SD 15secs)	1.000	0.859	0.069	WheelAngle (Range 3secs)	0.970	0.970	0.000
Lat_acceleration (Max* 15secs)	0.979	0.836	0.052	ACC_Abstandsindex (Mean 3secs)	0.963	0.962	0.000
WheelAngle (Range 3secs)	0.946	0.817	0.069	DTL_m (Min* 3secs)	0.951	0.951	0.000
WheelAngle (Mean 7secs)	0.940	0.800	0.059	DTL_m (Min* 7secs)	0.941	0.941	0.000
WheelAngle (Max* 3secs)	0.900	0.759	0.064	ACC_Abstandsindex (Max* 3secs)	0.940	0.940	0.000
Lat_acceleration (Max* 7secs)	0.882	0.750	0.051	ACC_Abstandsindex (Mean 15secs)	0.932	0.932	0.000
WheelAngle (SD 3secs)	0.944	0.722	0.076	WheelAngle (Mean 3secs)	0.928	0.928	0.000
Long_acceleration (Mean 15secs)	0.851	0.711	0.062	Ambient_Light (Min* 15secs)	0.928	0.928	0.000
ACC_Abstandsindex (Sum 15secs)	0.788	0.706	0.039	WheelAngle (Mean 7secs)	0.921	0.921	0.000
Long_acceleration (Max* 7secs)	0.871	0.674	0.064	Ambient_Light (Mean 15secs)	0.908	0.908	0.000
Long_acceleration (Range 7secs)	0.841	0.666	0.064	ACC_Abstandsindex (Mean 7secs)	0.889	0.889	0.000
Brake_pressure (Max* 15secs)	0.756	0.666	0.050	Temperature	0.871	0.871	0.000
Lat_acceleration (SD 7secs)	0.765	0.655	0.045	ACC_Abstandsindex (Min* 7secs)	0.859	0.859	0.000
ACC_Abstandsindex (Min* 15secs)	0.718	0.639	0.038	ACC_Abstandsindex	0.839	0.839	0.000
WheelAngle (Mean 3secs)	0.762	0.619	0.071	Ambient_Light (Mean 7secs)	0.839	0.839	0.000
Lat_acceleration (Mean 15secs)	0.707	0.618	0.036	Ambient_Light (Min* 7secs)	0.829	0.829	0.000
Long_acceleration (SD 7secs)	0.811	0.618	0.070	WheelAngle (Mean 15secs)	0.827	0.827	0.000

Table 4.9 shows the results obtained after performing the backward feature elimination procedure and presents the 15 most informative variables. Each row represents the prediction error obtained when removing the feature shown in the same row. The total number of features used in the model is also added in the first column. When it comes to the User Subjective, the error represents the proportion of wrong predictions from the total (1-accuracy) as a percentage. For the Response Time, the error is given as a function of the Mean Squared Error (MSE) [5].

Table 4.9: Backward Feature Elimination results.

User Subjective			Response Time		
Nr. of features	Error Rate	Removed feature	Nr. of features	MSE	Removed feature
15	0.355608	Lat_acceleration (Max* 7secs)	15	381.21	Day/Night
14	0.355628	Speed (Mean 7secs)	14	382.00	WheelAngle (SD 15secs)
13	0.356097	Long_acceleration (Range 15secs)	13	382.64	WheelAngle (Mean 7secs)
12	0.355836	Lat_acceleration (SD 7secs)	12	383.76	WheelAngle (Min* 3secs)
11	0.354996	Lat_acceleration (Max* 15secs)	11	384.21	DTL_m (Mean 15secs)
10	0.353362	Speed (Mean 15secs)	10	385.41	SpeedLimit
9	0.354632	ACC_Abstandsindex (Sum 15secs)	9	384.97	ACC_Abstandsindex (Sum 7secs)
8	0.355693	Speed (Min* 7secs)	8	386.34	Speed (Min* 7secs)
7	0.357876	Brake_pressure (Max* 15secs)	7	388.12	WheelAngle (Range 7secs)
6	0.363966	Long_acceleration (Max* 15secs)	6	391.46	Speed (Mean 15secs)
5	0.389109	ACC_Abstandsindex (Min* 15secs)	5	395.64	Ambient_Light (Min* 15secs)
4	0.419923	Day/Night	4	400.78	WheelAngle (Max* 3secs)
3	0.452912	SpeedLimit	3	405.87	RoadType
2	0.505651	RoadType	2	419.82	WheelAngle (Max* 15secs)
1	0.580095	WheelAngle (Max* 15secs)	1	441.87	ACC_Abstandsindex (Sum 15secs)
0	-	Speed (Min* 15secs)	0	-	Speed (Max* 7secs)

With regards to the User Subjective feedback, the most relevant variables found are: the minimum speed over the last 15 seconds; the maximum wheel angle over the last 15 seconds (which is expected to be correlated to bend scenarios); the road type (which is expected, since the route topology or available lanes are associated to this variable); the Day/Night parameter and the traffic indicator (ACC_Abstandsindex). It is possible to observe that the error varies very little when the number of features used is in a range between 6 and 15; this indicates that the model can be simplified minimizing the number of input variables without affecting its accuracy.

In relation to the Response Time feedback, a higher influence of the maximum speed over the last 7 seconds is observed. Other relevant variables are the traffic indicator (ACC_Abstandsindex), the road type and the wheel angle, which is expected since PDT buttons are located in the steering wheel and rotations may hinder the pressing of these buttons. In terms of Response Time feedback, the error does not vary very much when more than 8 features are used, which indicates that a model containing more than 8 features will increase unnecessary the complexity of the model.

It is evident from both feedbacks that the most informative variables are those associated with temporal evaluations over time; mostly in 7 or 15 seconds. These results highlight the importance of temporal variables over spontaneous values and are attributed to two factors: (1) the test procedure, in which participants may not reflect short changes in the driving complexity appreciated, and (2) the manner drivers understand the scenario and are affected by it.

Based on this analysis, driver may perceive driving complexity changes given a sequence of conditions rather than sporadic events. Surely, some sporadic events have a great impact

[5] MSE returns the average of the squares of the errors or deviations

on driving complexity, and as a consequence, fast and precise manoeuvres must be performed, e.g., the manoeuvre adopted to avoid an accident. But the process in which the mind handle these unexpected events is different since there is no time that allows the driver to adopt a new driving behaviour.

The identification of short changes in driving complexity is not the purpose of the estimator and other approaches can be taken for detecting these "sporadic events". Moreover, the predictor model is designed to be used by a manager of infotainment information, and these unexpected incidents should be controlled in a different manner; for example presenting alerts and restricting all entertainment functionalities.

4.2.3 Machine learning comparison

Different machine-learning methods were compared in terms of their accuracy and training time when using collected data. In order to gain an understanding of the computing required time, the machine used is the same employed in the data mining analysis (Section 3.3), it has a *i5* Intel processor and 8GB of RAM. The machine learning methods tested and its configurations are presented below:

- Decision Trees: This method is very useful since its resulting model contains a set of rules that are easy to understand and analyse. Decision Trees used in terms of User Subjective are configured by implementing a Gini index, Minimum Description Length (MDL) pruning, maximum depth of 12 and average split point. A regression scheme tree with a depth of 11 was employed for the Response Time.

- Random Forests: Combine different Decision Trees by selecting random features for the construction of each one. The model usually outperforms Decision Trees, since the prediction depends on more than one tree, although a drawback in the resulting model is that it is more complicated to understand and analyse. The Random Forest used for User Subjective is configured as follows: tree depth of 12, creation of 20 models and use of the Weka platform trough Knime. The depth of trees for the Response Time was limited to 11, and 50 models were created.

- Probabilistic Neural Networks: Consist of an evolution of Multi-Layer Perceptron; a PNN based on Dynamic Decay Adjustment (DDA) is used in this case. It creates a set of rules based on numerical data by adjusting a high dimensional Gaussian function depending on two thresholds (theta minus and theta plus). The PNN used in the comparison is configured using the following thresholds: theta minus equal to 0.2, theta plus equal to 0.4.

The different models where trained choosing the 6 most relevant features found after executing the dimensional reduction techniques previously exposed. The feature Speed (Mean 3 secs) was also included, because this variable could very useful for defining rules when the vehicle is stopped or when the speed exceeds the prescribed limit. The set of features included in the comparison for User Subjective were: Speed (Mean 3secs), Speed (Min* 15secs), Wheel Angle (Max*15 secs), Road Type, Speed Limit, Day-Night and ACC_Abstandsindex (Min*15 secs). In relation to Response Time, the variables used were: Speed (Mean 3secs), Speed (Max* 7secs), ACC_Abstandsindex (Sum 15secs), Wheel Angle (Max* 15secs), Road Type, Wheel Angle (Max* 3secs) and Ambient_Light (Min* 15secs).

Feedbacks collected during the test are not expected to be universally defined for each is scenario. User Subjective and Response Time are susceptible to many factors as personal appreciations, physiological responses, internal distraction during the test and other variables not measured.

In this regards, the different machine learning methods were evaluated using different proportion of training and validation dataset (80%-20% and 70%-30%). Datasets were obtained by taking a stratified and random sample of the complete dataset; this approach ensures that an equal proportion of each class is taken into account. Table 4.10 shows a comparison of the methods in terms of prediction accuracy when the feedback used was User Subjective and error (Mean Absolute Error)[6] when it regards to the Response Time feedback. The values presented in the table consist of the mean after executing 10 training and prediction iterations.

Table 4.10: Machine learning methods comparison (mean for 10 evaluation iterations).

Model Comparison		User Subjective		Response Time		
Machine Learning Method	Training-Validation Dataset [%]	Accuracy [%]	Training Time [secs]	MAE	R^2	Training Time [secs]
Decision Trees	80-20	62.81 ±0.24	31 ±1	459.1	0.305	5 ± 2
Decision Trees	70-30	62.73 ±0.27	27 ±1	465.7	0.293	4 ± 1
Random Forest	80-20	67.77 ±0.23	38 ±2	462.0	0.364	17 ± 1
Random Forest	70-30	67.75 ±0.24	31 ±2	467.7	0.354	17 ± 6
PNN	80-20	31.90 ±0.18	21210 ±520	1236.3	-1.795	440 ± 25
PNN	70-30	31.7 ±0.19	20350 ±465	958.3	-1.146	440 ± 20

Overall, training times for Decision Trees and Random Forest are low given the limited computational power used for creating the models. This is a positive indicator that even much more data could be added for training the model without requiring much time. Moreover, even some limited devices installed directly in vehicles could perform some operations in this regards. As an exception, PNN has shown to take much more time than other methods tested and therefore is not a probable good candidate for this purpose.

In terms of prediction accuracy and MAE, the method with the best performance is Random Forests, while that for training time is Decision Trees. The small difference in the training time arises from the number of models created in the Random Forests algorithm as compared with the single model created in a Decision Tree method. PNNs have proven to be the method with the worst performance for both metrics - training time and accuracy - which make them unsuitable for the dataset collected.

One may observe that the prediction accuracy does not vary significantly when training with one section of the dataset and validating with the rest. This validates the consistency of the data and the suitability of using some of the machine learning method tested for predicting unknown data. Furthermore, the training time is not very long, which means that more data can be added without the need for a high computing capacity.

As far as the fitness of the predictive model (R^2) for the Response Time is concerned, the model is found to have the best performance for both Decision Trees and Random Forests. In regards of PNN, negatives values of (R^2) are observed despite the larger training time; this shows that the model fits the data really poorly and worse than a horizontal line.

The Mean Absolute Error (MAE) for the best performing method in relation to Response Time, Random Forest, is in a range between 430 and 470 which is significantly high given the concentrated distribution of Responses time between 1100ms and 1200ms (see Figure 4.4). This distribution is a consequence of the small variance in response times

[6]The error in the fitness of a curve is usually measured in terms of Root Mean Square Error (RMSE) or Mean Absolute Error (MAE). Much discussion exists about which indicator is more informative about the deviation of the prediction. Overall, many researches chose MAE over RMSE to present their model evaluation statistics (Chai and Draxler (2014)).

of participants and to a smaller amount of data in comparison to that used for the User Subjective analysis (only the data around a PDT detection event are taken into account). Consequently, given the dataset size, Response Time feedback may be used for recognizing some more relevant variables, but not as a reliable indicator of driving complexity. Negatives values of (R^2) for PNN are also observed despite the larger training time; this shows that the model fits the data really poorly and worse than a horizontal line.

In order to measure the overall accuracy, confusion matrices offer a good representation of the dispersion of outputs in relation to expected values. Table 4.11 presents a confusion matrix arising from a Random Forest training using 80% of training and 20% of validation dataset. Overall accuracy is 67.91% and if the percentages of wrong outputs in the first and fourth quadrant (1-2 and 3-4 misses) are included, accuracy rises to 83.14%. Unmeasured individual differences regarding subjective perception probably account for these inaccuracies in the accuracy of the model.

Table 4.11: Confusion matrix of driving complexity estimation [%] (Random Forest)

	1	2	3	4
1	18,52	2,95	2,29	1,24
2	6,25	11,80	4,90	2,05
3	2,88	2,63	15,09	4,40
4	0,27	0,59	1,63	22,50

It is also observed that the percentage of wrong outputs summing cells 1-4 and 4-1 is only 1.52%, which shows that highly disperse perceptions are uncommon and an overall pattern exists in driving complexity categorization. For the purpose of the study, the accuracy obtained demonstrates the suitability of using some machine learning methods as an approximate indicator of driving complexity. The addition of some profile parameters in the future is expected to increase the model performance considerably.

Other metrics, such as sensitivity and specificity are also good indicators for comparing the performance of different models. If more data is collected, all these metrics must be also considered for selecting the machine learning method.

4.2.4 Individual pattern analysis

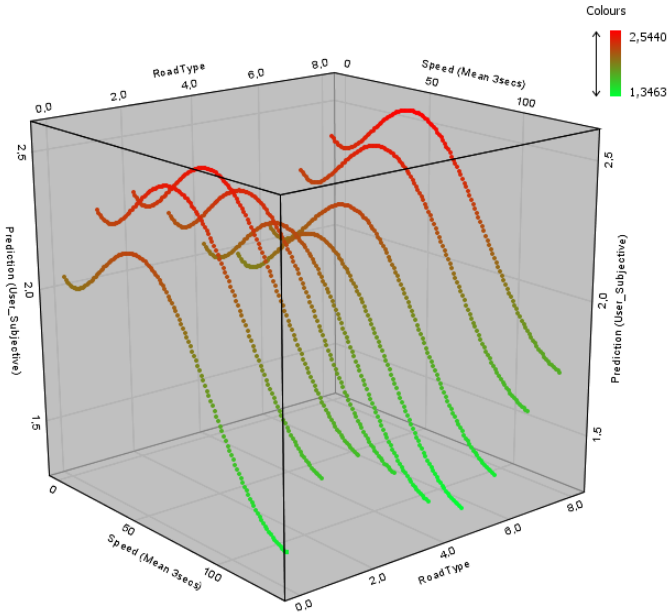
Additional analysis made in order to identify relations between variables revealed interesting patterns in the data. The first one is that the User Subjective feedback is not linked to specific input variables but rather to a temporal sequence of input; e.g., a single curve in a road usually does not affect the perceived driver complexity, although a sequence of curves probably will.

Likewise, some drivers changed their perceived complexity after travelling along the same type of road for some time; this could be attributed to a change in a variable not measured or to an adaptation of the driver to the scenario (Adaptive Attention). This is commonly noticed when joining to highways, where the new scenario affects driver perception, since the focus of attention must be shifted in order to interpret new stimuli. After some time, however, this complexity perception decreases and the driver feels capable of performing driving tasks adequately once again. The adaptation time is not studied in this research, but it indicates that the model output should not change often, but smoothly over time. Depending on the practical application, such as adaptive HMIs, such slow changes are mandatory because frequent changes in the interface will probably affect the driving experience and produce the opposite effect of increasing driver stress.

A curious finding was that participants tend to perceive a lower complexity when travelling at high speeds. These higher speeds are reached in motorways or highways with little traffic and few curves. Moreover, when participants travel at very low speeds they tend to perceive very diverse driving complexities; e.g., jam scenarios. These special cases are not considered by the model and can be managed by creating some additional rules that lead to a predefined categorization depending on the application.

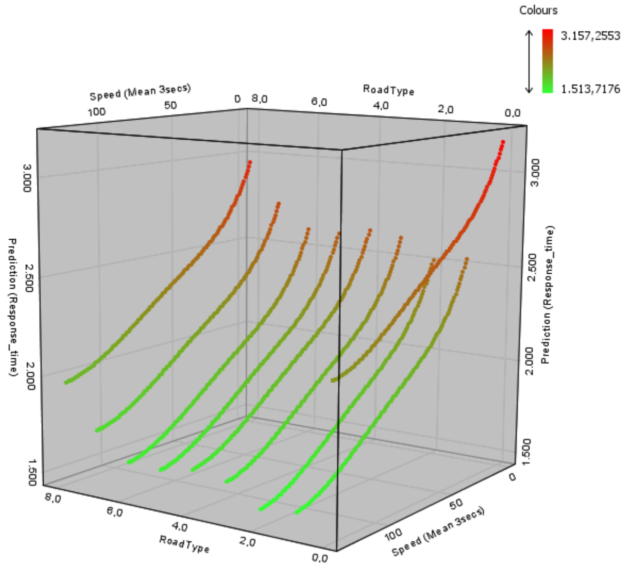
Figure 4.5 shows a polynomial regression (5th order) of User Subjective in terms of Road type and the moving average speed (3 seconds average). The graph, shows that User Subjective value increases when the speed is in a range between 40km/h and 70km/h and decreases at higher speeds. This result is a possible consequence of higher traffic levels or a lower confidence on route. Regarding road type, tertiary roads (values 7 and 8) are those with a higher impact on User Subjective, followed by motorway roads (values 1 and 2) and primary roads (values 3 and 4), while secondary roads (values 5 and 6) shows the lowest values.

Figure 4.5: User Subjective vs Speed(Mean 3secs) and RoadType.



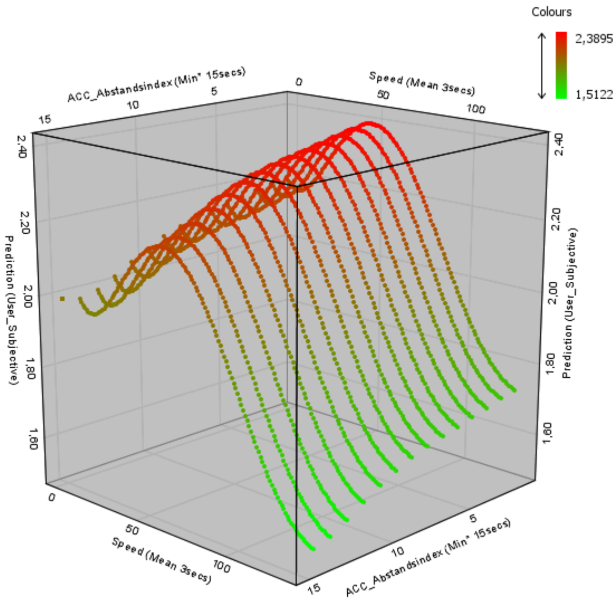
A similar result is obtained using the Response Time as comparison metric, see Figure 4.6. The graph shows a polynomial regression (5th order) of the Response Time in terms of the the type of road and the average speed in the last 3 seconds. Here, the Response Time is higher for unknown and tertiary roads (7 and 8 values) than for the rest. Also, for higher speeds the Response time tend to decrease, this result is consistent with the previously presented in Figure 4.5, when for higher speeds the perceived workload decreases.

Figure 4.6: Response Time [ms] vs Speed(Mean 3secs) and RoadType.



Interesting patterns were also identified in regards of the distance to the vehicle in front (ACC_ Abstandsindex), which can be seen as an indicator of the traffic level. Figure 4.7 shows how different values of ACC_Abstandsindex and Speed affect the User Subjective value after performing a polynomial regression (5th order). Here, smaller values of ACC, which represents a closer distance to vehicles and is an estimation of higher level of traffic, are associated to a higher value User Subjective.

Figure 4.7: User Subjective regression in terms of Speed(Mean 3secs) and ACC_ Abstandsindex (Min *15 secs).



4.2.5 Model validation

The performance of model is evaluated in a real driving scenario; this evaluation is based on the perception of an expert in road safety. The model is based on a Random Forest machine learning method and uses the same previously evaluated dataset. The training of the model is performed by employing the same 6 variables as those used in the machine learning comparison for User Subjective feedback (4.2.3), while integration in the vehicle is done by developing a Java program with the model loaded.

A Java program containing a large set of libraries was used a single container for the driving complexity estimator. This program was designed to access needed CAN variables, filter and adapt incoming variables, make previous computations such as determining the road type from a preloaded cartography, and use the created model to predict the driving scenario in real time.

All the tools employed are Open Source and multi platform which allows an easy integration in other vehicle hardware with low investment. The final model is packaged and loaded using the Java Weka Module v3, the final resulting size of the model was just about 12MB which is small enough to be loaded current vehicle hardware.

As regards the model performance, a fast calculation time is observed, which makes it suitable for real time operation. An accuracy of 56.1% is obtained for exact matches and 81.49% when combining 1-2 and 3-4 misses. As an occasionally noisy and variable output was found as a consequence of sporadic variances in estimations over time, the model was re-adjusted to stabilize it. The procedure consisted in taking the mean of the last 5 seconds estimated instead of the direct value. In this manner is guarantee a more smooth and stable change over time.

The accuracy obtained with the stabilized model reached to 57.95% for exact matches and 82.77% when combining 1-2 and 3-4 misses. The confusion matrix associated to these results is shown in Table 4.12. Here, a small amount of incorrect outputs in cells 1-4 and 4-1 (0.73%) can be seen, which is a good indicator of the small deviation in estimations.

Table 4.12: Expert validation confusion matrix [%]

	1	2	3	4
1	8.26	1.59	0.30	0.25
2	15.37	13.08	7.15	2.28
3	0.69	5.01	10.90	3.75
4	0.48	1.07	4.12	25.71

Figure 4.8 shows the route used as reference in this validation stage; each colour represents the driving complexities perceived by the driver, ranging from 1 (very low) to 4 (very high). The first illustration corresponds to the expert's own perception; the second is the given model output, and the third is the smoothed model. A discrepancy between original and estimated values in regards to category 1 (blue) and category 2 (green) is observed in part of the trip. As previously explained, this differences are expected to be a consequence of the different perception of risk and is acceptable for the purpose of the procedure.

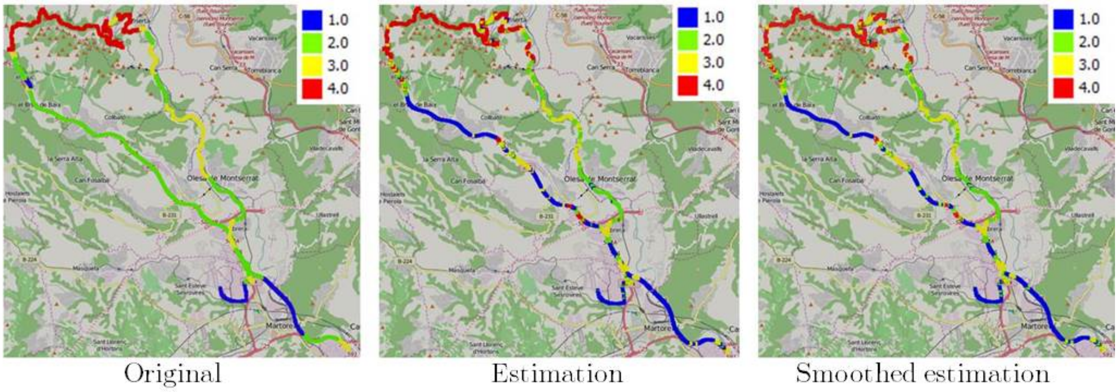


Figure 4.8: Expert driving route.

4.3 Limitations

This section presents some limitations of the study that could be overcame in future research works. First, since the main purpose of the study was to create an estimator of driving complexity that could be incorporated into current vehicles with a low investment, not all the variables that may have an influence in driving complexity were evaluated. Moreover, the output value was expected to be used as a reference for managing the flow of information in infotainment systems, and for this purpose, a high accurate model is not needed. Driving scenarios are heterogeneous and a model incorporating more variables as inputs would probably increase it accuracy.

Furthermore, a larger sample of participants may increase the accuracy of the created model and allow a profile characterization; for the purpose of this study the sample was limited due to time and costs constrains. Therefore, participants were selected according to homogeneous parameters in terms of driving experience and driver periodicity, thereby reducing the effect of profile patterns.

The perceived driver complexity of a road may also vary according to user expectation or to the unpredictability of a road. The attention level is therefore expected to be higher when driving on unfamiliar roads than on more familiar roads. This parameter was not evaluated since the majority of routes where already known.

Analysis of the trips revealed interesting patterns where perception reported by participants were not linked to sporadic events but rather to a response of a sequence of input; e.g., participants did not use to report changes due to a single curve in a road, although they usually did in sequence of curves. Likewise, some drivers changed their perceived complexity after travelling along the same type of road for some time; this could be attributed to changes in unmeasured variables or to an adaptation of the driver to the scenario. For some practical applications, such as adaptive HMIs, slow changes are mandatory, since a very dynamic the interface will probably affect the driving experience and produce the opposite effect of increasing driver stress.

Autonomous driving assistants are becoming increasingly common. These mechanisms may alter the driving task, making driver performance invisible to the workload estimator. The impact of the ACC on driving complexity was not evaluated in this test, since trips with this data were not taken into account. However, future tests could make a significant contribution in this regard.

4.4 Summary

This chapter exposes the procedure followed to create a model capable of estimating driving complexity in real time. For creating the model, data were collected from several variables in real driving scenario and two variable were considered as feedbacks for identifying patterns: drivers' own perception and their variability in response time to a PDT. The study shows the suitability of implementing a system that estimates driving complexity without the need for high investment. No additional hardware is required for the system to be introduced into vehicles, since the input variables are already accessible and the required computing power for making predictions is significantly low.

The procedure followed for creating the model consisted in: the application of several dimensionality techniques mainly focused in simplifying data collected, and the evaluation of different machine learning methods in regards of finding the most suitable for the purpose followed. On this understanding, an estimation model enable the categorization not only of scenarios that are already recognized, but also unknown circumstances based on previously collected data.

Among the different methods evaluated, Random Forest was the model with the best performance for the dataset collected. This model was based only in one of the two feedbacks considered: the User Subjective perception of participants. Variability in response times were small and a concentrated distribution of values in a small range was observed. In consequence, this feedback is useful for determining relevant variables in order to categorize driving complexity, but is not suitable for training a predictive model because of the low small variability.

The higher concentration of response times in a small range is an indication that drivers tend to have similar response times. Deviations may be associated with higher driving complexities, or in some cases with failures to check the PDT. A larger amount of data may overcome this issue and the feedback could be implemented as a second validation value.

In regards of User Subjective, results show that driver perception may be a good indicator of driving complexity, since the responses of the different participants prove to be similar. Some minor variations in the perceived complexity exist between participants due to the risk perceptions of each individual. This dispersion could be minimized by considering profile features as an input to the system. Nevertheless, depending on the application, these dispersions are not relevant.

Future works can include profile characterizations as an input and in extending the length of the test in order to take more scenarios into account. Moreover, an online service for gathering drivers' data can be used to create maps of driving complexities. The predictive machine could also learn from road safety experts rather than regular drivers in order to establish more standardized and regulated driving rules.

The complexity value may also be useful for executing reconfiguration in some Autonomous Driver Systems by deactivating or adjusting some thresholds. An alert system can also be created to inform drivers about a greater scenario complexity as well as ensuring complete awareness.

Chapter 5

Adaptive User Interface

This chapter exposes the procedure followed for developing an Adaptive Human Machine Interface (A-HMI). The proposed system was designed to manage the presentation of information and restricts drivers' interaction in accordance with the driving complexity. All this, with the aim of increasing road safety and ensure a proper level of driving experience. As indicator of driving complexity, the model previously presented in Chapter 4 is employed.

Current design principles for IVIS offer scarce information for creating adaptive or dynamic infotainment systems that modify the interface design depending on certain circumstances. In this regards, the initial purpose of the study is to propose a set of design principles that take into account the driving complexity level for adapting the behaviour of the HMI. Once these principles are defined, these can be used as reference for creating the A-HMI, and evaluating the real benefits of the system.

The proposed A-HMI is based on an already commercial available Head Unit belonging to the brand SEAT. This strategy allows to compare the real impact of the proposed system on drivers perception in relation to an already validated system. A test designed for evaluating participants perceptions under real driving scenarios was performed. In this test, metrics evaluated were drivers' acceptance, road safety perception and drivers' experience.

5.1 Design Principles

Based on available principles and guidelines, cited in Section 2.1.2, an adapted set of recommendations is proposed for creating adaptive interfaces. This set of principles are classified in 5 main groups according to overall concepts: Consistency, Modality, Adequate timing, Situation Awareness and Representation of information.

5.1.1 Consistency

Consistency is defined as an "agreement or harmony of parts or features to one another or a whole" (Merriam-Webster, 2004). In a HMI context, it can be expressed as a common design pattern for the different elements that constitute the user interface. This concept has been deeply study and advised by several recommendation guidelines such as: ESoP (Commission of the European Communities, 2008), where is stated that the information should be displayed in a uniform manner and be consistent among available interfaces, exposed by Shneiderman (Shneiderman, 2010) where his first principle states "Strive for consistency" and by Nielsen (Nielsen, 1993) where the fourth principle of his work exposes the importance of "Consistency and standards".

This principle is one of the most relevant when designing interfaces since it is very linked to users' expectations when interacting with the system; any interface must guarantee to the users an easy understanding when browsing between different screens and menus. This can be achieved by creating a guideline design and be sure to apply it on every created screen or interaction mode. A high consistency helps the user to understand the overall meaning of icons and terminologies employed. Moreover, the learning time needed for a achieving a proper management of the system can be reduced considerably and allow users to perform actions faster.

In order to achieve a high level of consistency, elements must be incorporated following a same structure and design pattern. Some elements that need to be standardize are: menus, terminology, help screens, icons and colours. Also a common interactive flow between different context must be guaranteed; as an example, if the button that permit the returning to a previous screen is placed in the top, this arrangement must be maintained for every screen that includes a back button. Moreover, the button should be graphically represented equally every time.

In the same way, basic graphic elements used to represent certain information such as lists, grid menu, dropdowns, must be coherent and be used to show similar information; for example, a list with system settings could be shown as a list or as a grid menu as long as this representation scheme is used in all sections of system settings. For this, the key is to define and respect a design guideline in which the generic elements that will be used throughout the application are well defined and planned to contain certain kind of information.

On the other hand, it is clear that some space should be left to the creation of certain unique screens that may include non-generic elements in order to highlight information or emphasize the importance of the screen among the rest. Some examples of use can be the creation of a main screen, representation of elements in a map or interactive screens with graphics.

With regards to an adaptive interfaces, the concept goes beyond the representation of information in static screens. Here, the same adaptation procedure must be undertaken for the different driving complexity scenarios. As an example, if a button associated to a functionality is removed in one screen when certain driving complexity is reached; this button must be also removed of any other screen that have the same functionality. Some exceptions may occur in case that some functionality offers a mandatory action for certain context.

Overall, the strategy for showing-hiding certain information represented in long texts, interactive lists, images, animations must be clearly defined and coherent among all the interfaces. When creating adaptive interfaces, this strategy must be included as part of the design guideline and be provided to designers and developers in the initial phases of the project to ensure that the application is planned from the beginning following this set of rules.

5.1.2 Modality

In a driver-system interaction, information can be transmitted –driver to system– and received –system to driver– through different channels. When transmitting information, actions are initiated voluntarily by the driver employing mechanisms such as air gestures, touchable gestures, push of buttons and voice commands. In the reception, information is generated by the system after an user request or directly when the system decides to present it. Reception channels available for drivers are the biological human senses such as hearing, sight and touch.

The modality principle explores the benefits and disadvantages of using different channels for both transmitting and receiving information. A good IVIS should be capable of management the flow of information trough different channels regarding: the type of interaction (given by its duration, complexity and frequency of use), its purpose (relevance), its priority, the existing interference in used channels and users preference. Depending on all these characteristics, different repercussions on road safety and user experience are given.

A large amount of literature exists in regards of multi-modal interfaces; a very good compilation of researches can be found in the book "The Multisensory Driver" (Ho and Spence, 2012), here the benefits of using different modality channels are explored based in conclusions given by previous researches. The book uses as baseline the Wicken's Multiple Resource Theory, which states that the performance of an activity is affected if there is interference in the same channels needed for performing that task (see Section 2.2.3.

Among the transmission channels –driver to system–, visual interfaces are one of the most used mechanism to present information. It offers numerous advantages since a large amount of information can be presented simultaneously, and also it can be simplified by employing simple icons. Once users understand the given design using a visual interface, they usually become agile interacting with it and are capable of recognizing information with just a few glances. This is the case when looking at simple messages as fuel level, warnings and alerts. Moreover, this kind interfaces can offer a unique look and feel that distinguish each HMI design from the rest. Also, user can decide weather or not to visualize it.

A drawback of visual interfaces is that they are not recommended for representing priority information. Some justifications are: a faster response time achieved using auditory or tactile stimulus, the more alerting nature of other mechanisms, the need of periodic glances to detect the information, and also the higher interference with the driving task, which is mainly a visual tasks.

In this regards, tactile and auditory channels have shown to be adequate for most priority messages. Signals transmitted trough these channels are detected faster than visual stimulus and it is not need that the driver divert the look from the road. Finger response time for different stimuli (Visual, Auditory and Tactile) were compared by Ng and Chan (Ng and Chan, 2012). Results showed that the time in response to the tactile stimuli was significantly shorter, followed by the auditory stimuli and then the visual stimuli. Other studies show than in average, the response time of a person to a light stimulus is larger for an auditory stimulus; 190ms vs 160ms according to (Daiss-fechner and Boat, 2005) and 289ms vs 236ms according to (Kosinski, 2008).

Auditory signals are usually transmitted using a simple combination of tones or a verbal message; the implemented scheme must be careful planed depending on the functionality purpose since unclear notifications may lead to confusion. The use of spoken auditory messages is suggested for many applications since it carry more information than simple tones (Ho and Spence, 2012). These analysis report the benefits of using verbal cues obtaining as result than participants were able to comprehend easier their meanings than for single tones and direct their spatial attention properly. As a drawback, a person's response to a speech warning may be slower in a emergency situation, specially if the operator does not understand the meaning of the speech until it has finished.

In this regards, a simple combination of tones may seem to be appropriate in scenarios where a fast response time is required. Nevertheless, this kind of scenarios are uncommon and the user will not probably understand the meaning of tones announced until he has checked in a visual interface the reason of that alarm. Some applications in which this approach shows a higher acceptance are the indication of obstacles when parking or an incoming call.

An important consideration to take into account when using the auditory channel, is that warning messages must not be obscured by entertainment auditory sounds as music or not priority notifications. For this purpose a priority scheme that order the information presented is recommended. Also, the system should allow the user to deactivate the notification of non-priority messages trough auditory channels. An approach employed in previous studies for adding more informative value to auditory messages consist in sending information trough different auditive source spatially distributed (Ho and Spence, 2012).

The same study also reports the benefits of incorporating tactile interfaces into vehicles, its main advantage is related to a faster detection time in comparison to auditory and visual stimulus. An experiment for evaluating this interface consisted in comparing the response time of 16 participants for detecting a critical visual event just trough the visual channel or accompanied with a tactile stimulus. Results show that participants responded both more rapidly and accurately when the critical visual event was preceded by a vibrotactile cue.

Auditory and tactile interfaces have proven to be well suited to announce critical situations and are helpful for reducing the deviation time of the eyes from the road. In order to improve the understanding of these types of messages, the addition of redundant information through a visual interface is recommended in order to help the driver confirm the understanding of the message. Visual information allows to present more information at the same time, provides more creative designs and is not annoyance for the driver compared to auditory or tactile messages.

In regards of the transmission channel –driver to system–, a higher flexibility in the design is allowed. These interactions are activated by the user and most of the design is oriented to offer a higher level of user experience that does not jeopardize road safety. In order to guarantee safety, system controls should be designed such that they can be operated without adverse impact on the primary driving controls (Commission of the European Communities, 2008).

The approaches followed by vehicle manufacturers is oriented to ease the control of the system integrating more accessible interfaces such as: buttons in the steering wheel, voice recognizers and gesture detectors. Depending on the applications some ways of interacting are preferable; for example, the initiation of a simple actions as "go home" should be provided with the lowest simplicity, here gestures or voice commands could be advantageous. Other cases as the browsing of albums or artist may be more annoyance to be controlled by voice/gestures for certain users.

Regarding safety, three different music retrieval systems (one with a multiple entry touch interface, the iPod^{TM}, one with a multiple turn voice interface, and one with a single turn voice interface) where evaluated in terms of secondary task performance, eye behaviour, vehicle control, and workload while participants were driving in a simulator. Results showed that when using voice interfaces the total time drivers spent with their eyes off the forward roadway was reduced and that the multiple turn voice interface significantly increased the workload (Garay-Vega et al., 2010).

Grane evaluates the impact on driving performance when using an in-vehicle haptic rotary device for interacting with a visual display depending on four modes of operation: visual only, visual + partly haptic, visual + haptic, and haptic. Results show that the full haptic support had the least negative effect on driver performance (Grane and Bengtsson, 2013).

A combinations of modalities for offering the same functionality is also a good approach. A comparison of the effect of interacting with a (1) single visual head-up or (2) an audio and a multi-modal display obtained as result that the interaction with the second is faster and more efficient (Jakus et al., 2015).

In an adaptive HMI, modality can be adapted reducing the amount of redundant interfaces that can be used to interact with the system. The proposal scheme restricts those interfaces more complicated to manage that provide a better user experience but may jeopardize safety. For example, the interaction with the system using a touchable interface can be limited to be activated using only voice commands.

Subjective parameters should also be taken into account when designing the system; Kong proposed an automatic modality adaptation depending on user preferences (human-centric adaptation). Results show that each user has individual preferences of how to interact with the application (Kong et al., 2011).

5.1.3 Adequate timing

Existing guidelines make two main recommendations in regards of adequate timing; one is oriented to ensure a timely feedback after the user performs an action, the other is stated by Schneiderman in one of its principles “short-term memory load”. He exposes that humans have a limited capacity to process information and therefore the sequence of steps needed for performing an action should be kept to the minimum possible (Shneiderman, 2010).

To comply with this principle it is mandatory to avoid the need for long sequences of interactions to execute an action; for example, when changing a setting in a system it must be avoided the need of multiple touch events or the navigation through different screens to apply that change. As an alternative, different generic elements such as list, dropdowns and sub-pages can be used to minimize the interaction.

In regards of the adaptive HMI, several adaptations can be performed when the driving complexity increases. A common approach consists in implementing schedulers that delay notifications depending on certain driving parameters. Based on previous researches (see Section 2.3, these delays are proposed to be given as a function of the following parameters: importance of the notification, current driver workload, given scenario, estimated driver workload given the additional functionality and user preferences.

Under this strategy, not all the notifications are delayed for the same level of driving complexity but a group of these based on some criteria. For example, an incoming call which is expected to have a lesser effect on workload than an visual incoming information about some weather forecast can be allowed in higher values of driving complexity. This delay should be computed taking into account user preferences and also some alternative mechanisms for avoiding a degradation of user experience should be provided. As an example, incoming calls should not be delayed for much time since the user may be expecting an important call. Instead the system can make a recommendation to the user for sending a receipt message such as: “Call you later”.

Also, adaptive HMI changes must be properly managed depending on the user interaction with the system at that moment; supposing the user is interacting with the system just when the driving complexity changes it must e carefully evaluated if apply that change immediately or wait for a few seconds to avoid user confusion. This applies both to cases of increase and decrease in the degree of complexity of the route.

In addition, adaptive changes should not be very frequent, but should be applied after the driving complexity has stabilized at a certain level. This is mandatory to avoid very frequent changes that can distract and confuse the driver instead of helping him to interact with the system. The time required to ensure that the complexity of driving has been stabilized will depend on how many levels of driving complexity are used and it is advisable to perform usability / distraction tests to identify the most appropriate value.

In low driving complexity scenarios, some safety related measures can also be taken. Based on the Waard's model (see Section 2.2.3, a low demanding scenario affects in a similar manner the driver performance as a complex one. Therefore, a countermeasure could be the addition of alerts or more dynamic interfaces when the driving complexity has been low for a long period of time.

5.1.4 Situation Awareness

The term "Situation Awareness" is referred to the user understanding of its surrounding environment and the consequences given by its actions. Much literature brings out the importance of this concept when designing new user devices or interfaces. Stanton exposed that one of the problem of modern vehicles is that are designed to isolate the driver of the external environment in regards of increasing comfort. This isolation reduces the situation awareness level of driver and therefore can have and indirect impact on road safety (Stanton et al., 2015).

When designing HMI, a proper level of SA is present when the user is always conscious of the purpose of the current interface; understand how to access other contexts and have an overall idea about what will happen when an interaction is started (Gruhn, 2011), all this without being distracted too much and be isolated from the driving task. Nielsen states two principles related to SA: (1) the system should always keep users informed about what is going on through appropriate feedback, and (2) the system should make use of language and concepts familiar to the user, rather than system-oriented terms (Nielsen, 1993).

This can be achieved by creating simple user interfaces with standardised structures that always show the user what type of screen is shown, what context is referred to, and have clear indications of how can be returned to a previous screen or main menu. Isolation from primary driving tasks can be decreased reducing the number of steps needed to execute an action, representing clear all the information, summarizing in single screens related contexts and reducing periodic animations that may not add relevant information to the user.

With regards to an Adaptive HMI, design changes should be introduced smoothly and an explanation must be given to the user about this dynamic behaviour. An indication of the current level of driving complexity is proposed to be presented in order to help the user understand the current scenario. This indicator can help the driver realize the reason of a change in the interface. Moreover, after the user is familiarized with the system, he may known what to expect under certain conditions and recognize the limitations in that moment.

This indicator can also helps the user realize that he should take more attention to the road and also perform some countermeasures as reducing speed and increasing the distance to the next vehicle.

5.1.5 Representation of information

This principle is the main core of the creative part in design and many concepts are associated to it such as understandability, coherence, look-and-feel and simplicity. A good design should guarantee a proper user experience offering not only an attractive appearance but also a high informative level.

An adequate representation of the information does not only seek the correct creation of individual and specific screens for displaying some data, but it is also a necessary principle to be taken into account during the first steps of the user interface development. This principle is mandatory for defining the correct graphical elements needed to show information throughout the functional flow of the application. For example, the management of an error message could be represented by using a popup, a single icon change or some text; this decision which depends on the priority of the message must be defined following an standardised strategy.

Nielsen expose three usability principles related to the representation of information that can be summarized in: (1) interfaces should provide an accessible and understandable mechanism to abort actions and go back in each context, (2) design should be focused in preventing errors to happen instead than in offering good error messages, and (3) design should be aesthetic and minimalist avoiding the addition of unnecessary information (Nielsen, 1993).

The *Eight Golden Rules* also give recommendations regards of representation of information. It proposes to enable shortcuts that reduce the number of interactions needed to perform actions, specially if these are executed commonly. The relevance of providing informative feedback is also highlighted, these responses can be modest for minor actions and substantial for infrequent and major actions. Additionally, easy reversal actions must be accessible in order to relieves user anxiety in case of errors, as well to encourage the exploration of unfamiliar options (Shneiderman, 2010).

Besides usability, enjoyment is a relevant metric that vehicle manufacturers consider when designing new systems. A unique personalization with new features and creative design styles allows a brand to differentiate from others. It is during this phase that is important to establish the boundaries of such creative process in regards of road safety before the design advance forward.

In an adaptive HMI, when the driving complexity increases, the graphical representation should be reduced from a more creative nature to a more conservative one if it assures a higher level of safety. In this regards, some approaches can be the reduction of information shown, the increment in size of relevant buttons or information, and the offering of a better contrast and selection of colours that ease the recognition of information. Also, very dynamic animations that may distract the driver or delay the execution of an actions should be avoided.

5.2 Adaptive System Integration

The procedure followed by vehicle manufacturers in the design and development of infotainment systems is quite complex. Many tasks and validations must be carried out before the system is considered adequate for production. Given this slow process, a prototyping phase that allows performing some initial usability and acceptance test is recommended. Under this approach, new functionalities or designs can be evaluated without delaying the initial schedule.

The main goal of the prototyping phase is to assess the benefits of the functionality and design liking from the user's perspective. For this, neither the internal logic of each functionality, nor the real processing of actions need to be programmed. On the contrary, this phase should be completed as quickly as possible since some of the results obtained here will used as reference for creating the final application. Once the prototype has been designed, usability tests using external participants are encouraged, their feedback is crucial for getting notice of design failures that can be improved.

During this project, the prototype system consisted of an instrumented vehicle containing a Linux machine. The machine provides access to almost every variable in the vehicle and to user interfaces as the Head Unit and the Kombi. Applications and functionalities can be directly programmed using external software since, the machine offers screen replication services that give access to the user interfaces. In a similar manner, services that monitor user interactions with the interfaces can also be integrated.

A framework provided by Volkswagen was used for creating the Adaptive HMI. This framework, based on Open Services Gateway initiative (OSGI), launches a server called HMI Server that replicates some of the functionalities available in the infotainment system. In a HMI Server, each application or process, programmed in Java, is managed independently in a separated thread (bundle). This approach allows the launching or termination of processes without affecting the execution environment. For the programming of graphical interface, the modelling is achieved using JavaFx, a software platform for creating desktop applications and Rich Internet Applications (RIAs).

5.2.1 HMI Server

The HMI Server responsible of rendering the user interface is based on OSGI, this framework provides a modular structure in which applications or services can be installed, started, stopped, updated, and uninstalled without requiring a reboot of the server. Thanks to this approach, if a component is stopped, dependent services are also stopped without affecting the rest of the system. Also, this kind of systems are very stables, scalable and suitable for production environment.

In a OSGI structure, independent components are called bundles. Bundles are composed by a group of Java classes with additional manifest headers used to indicate exported services and permissions of use. Services are planned to create abstract methods that can be accessed by different contexts or applications. Any bundle can implement service interfaces and register itself to a main Service Registry. Any allowed client can get a list of available services and subscribe to one of them for reacting to it when it starts, stops or publish a notification.

The OSGI framework is divided in the following layers: Security Layer, Module Layer, Life Cycle Layer, Service Layer and Actual Services. For the purpose followed, it is of special attention the Life Cycle Layer which provides an API useful for controlling the operations of bundles. This layer defines the state of a bundle which can be: installed, resolved, starting, active, stopping, inactive or uninstalled (Osgi, 2011). A bundle can only be active if all the required components (other bundles or services) are already active. The platform provides mechanism to automatically start required bundles when needed.

Currently, there are three popular open source OSGi containers: Equinox, Knopflerfish and Apache Felix. HMI Server make use of Equinox 3.8.2 which is a Eclipse IDE modular OSGI container. Equinox allows to configure virtual machine capacities, set the order in which bundles should be activated and provide logging tools for monitoring the state of each one. Together with this framework, the graphical interface was developed using *JavaFx 8*, a platform designed to ease the development of GUIs. This platform provides a set of Java libraries that allow a fast creation of graphical elements to be integrated in applications.

The HMI Server consists on several background services and applications. Each main application, also called Context, is defined as a unique bundle. Contexts are in charge of rendering the graphical interface and providing infotainment functionalities thanks to the access to external services. Most of these external services provide communications mechanism with the vehicle trough protocols such as EXLAP or CAN. Others processes available are: media player, image loading, phone commands, string parsing and vehicle location. Table 5.1 shows the main bundles defined in the adaptive system.

Table 5.1: Components of the adaptive HMI Server.

Type	Name	Function
Components & Services	infotainment.audio	Audio player
	infotainment.media.fx	Gallery loader for Artworks (audio files)
	abt.input.simulator	ABT buttons simulator
	settings	MIB overall settings
	CANBAP_Connector	Communication with vehicle (Access to variables)
	RobotMIB	Notification of Head Unit touchscreen and buttons interaction
	Navigation_Control	Navigation operations (Guidance)
	EXLAP_Service	EXLAP communication service
Contexts (Applications)	Dynamic.mib.mib	MIB main context (Controls other contexts)
	Dynamichmi.mib.media	Media context
	Dynamichmi.mib.tuner	Tuner context
	Dynamichmi.mib.gallery	Gallery context
	Dynamichmi.mib.navigation	Navigation context
	Dynamichmi.mib.car	Car context
	Dynamichmi.mib.setting	Setting context
	Dynamichmi.mib.phone	Phone context
Estimator	Workload_estimator	Estimate driving complexity and notify value.

The adaptation of the interface is achieved by modifying the skin or appearance of each context depending on the driving complexity value. For this purpose, a very useful tool in JavaFx called "Bindings" is used. Binding, which are based in JavaBeans component architecture, allow defining *"Listeners"* and *"ObservableValues"*. When an *"ObservableValues"* changes, those *"Listeners"* subscribed are notified and specific actions can be started; in this case the adaptation of the interface.

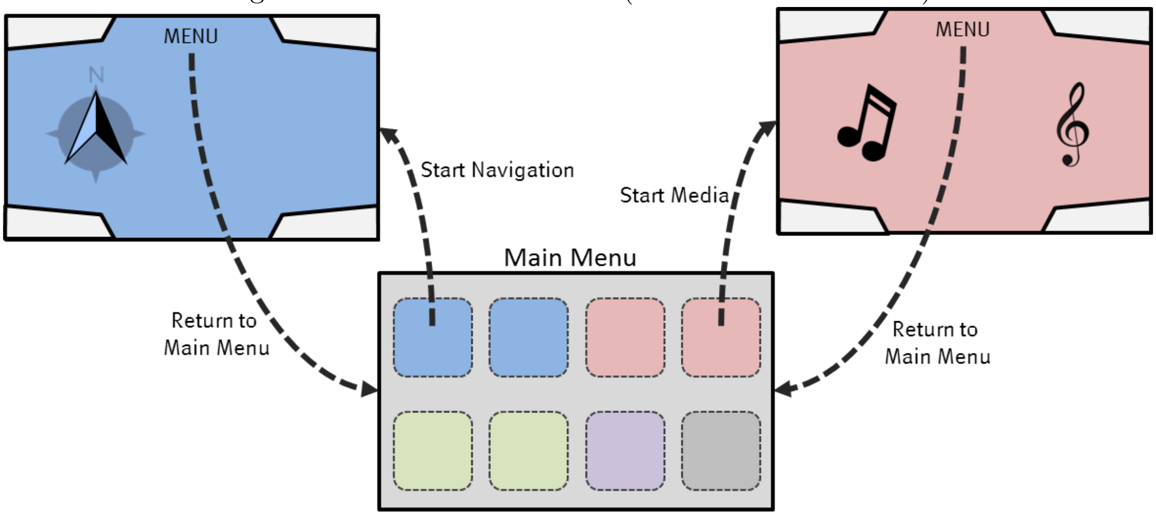
The strategy followed, consisted in defining a global property, *"IntegerProperty"*, that stores the driving complexity value and a *"Listener"* for each context. Any time the driving complexity value is updated, each context gets notified and specific process are started to execute the adequate HMI changes. Transitions between design changes were smoothed employing JavaFx tools as *Transitions* and *Timeline Animations* (Dea et al., 2014).

The service in charge of notifying this complexity change is incorporated in the Workload_estimator bundle. The service implements the estimation model developed in Chapter 4. More details about the structure of the server and code used for creating the application is exposed in the Appendix Section F.

5.2.2 Adaptive design implemented

The Head Unit used as baseline contains a main menu that provides the user with easy access to diverse applications. Each application has a very accessible button to ease the returning to the main menu as it can be seen in Figure 5.1, most of the contexts have also a button in each corner associated to an action. The adaptive changes are defined independently in each context, but comply with the important principle of consistency in changes so that each context is adapted in the same degree. Following the same principle, similar applications have same base colours in their design, e.g., media player and radio contexts use the colour red in common graphic representations as icon borders, while setting contexts employ yellow colours.

Figure 5.1: Baseline HMI used (main menu interaction)



Also, an indicator of current driving complexity was added in every display to inform the driver about expected changes (Situation Awareness principle). The indicator was represented by a simple icon showing a iconographic face representing different moods: smiley face for low driving complexities and sad or worry faces for higher states of driving complexity. This icon was always located in the same position of the screen inside the different contexts (upper right part of each display). Regarding the design, the icon could be later replaced by any other indicator as for example, a different colour contrast in the background, or the change of the ambient light of the vehicle.

The proposed HMI system adapts the interface according to four levels of driving complexity. Thus, as the level of complexity increases, the aim of the system is to ease driver interaction and thereby assure better road safety. A commonly employed and very useful strategy in adaptive systems is the implementation of a scheduler that delays the presentation of information depending on road conditions. However, in the current study is not implemented an scheduler since this topic has been already studied extensively and validated as adequate.

The adaptive nature proposed is characterized by the implementation of two major design alterations; restrictions, focused on restricting some functionalities in complex scenarios; and appearance, focused on the re-organization of elements regarding how the information is displayed. Greater and lesser design alterations were implemented in order to evaluate user acceptance and appreciation to different kind of changes in this type of system.

Restrictions are applied during complex driving scenarios and on functionalities that are not commonly used or require a higher level of attention. To this end, functionalities are first classified according to three parameters: role inside the application; relevance to the user, and interaction complexity. Interaction complexity is objectively measurable and can be analysed with variables as glance time or the number of clicks required to complete the task. For example, some functionalities involving browsing through lists or writing messages using a keyword should be characterized as being of higher demand than functionalities such as clicking a "next" button to change a track or a song.

The classification of functionalities must be carefully planned, since the impact on usability and user experience will be directly linked to whatever restrictions are applied. Some non-relevant functionalities suitable for restriction in complex scenarios are the search for a particular track on an album, the selection of POIs in the navigation or the saving of new presets as favourites on the radio.

The alteration in appearance eases the interaction with commonly used, low-complexity functionalities inside each application. The appearance of the design in high complex scenarios is more focused on usability and providing easy access to information rather than the look or aesthetics of it. Some of the incorporated changes could highlight the relevance of commonly used functions such as "next track", "play", "pause" or "stop-start". Other modifications could be made to hide non-relevant information; for example, as regards Navigation, areas outside the current route could be hidden in order to make map reading easier.

Table 5.2 shows a classification example of the some functionalities in terms purpose inside the application; relevance to the user, and interaction complexity. Each functionality can be initiated by the user when he demands some actions to be performed, or directly by the system when notifications or information are presented to the user. The parameter "Relevance to the user" is subjective and depends on individual preference, therefore the information presented in the table is added as an example. From this table, the adaptation in the design should be based initially on the "Interaction complexity" since it represents, in some way, safety implications and "Relevance to user" given that guaranteeing a good user experience is one of the premises of the proposal.

Table 5.2: Classification of functionalities.

Functionality	Context	Initiated by	Classification [Low (1) to High (3)]		
			Role in application	Relevance to user	Interaction complexity
Inform low pressure in tires	Alarms/Notifications	System	3	3	1
Inform windscreen liquid empty	Alarms/Notifications	System	1	2	1
Alarm about fuel level	Alarms/Notifications	System	2	3	1
Volume up-down	Infotainment	User	1	2	1
Mute/Unmute	Infotainment	User	1	1	1
Listen to music	Infotainment	User	2	3	1
Next-Previous song	Media	User	2	3	2
Change audio source	Media	User	2	1	3
Adjust equalizer	Media	User	1	1	3
Search by album/song/artist	Media	User	2	3	3
Set destination (favourite)	Navigation	User	2	2	1
Set new destination (text writing)	Navigation	User	2	3	3
Stop navigation	Navigation	User	3	1	1
Show arriving time	Navigation	System	1	3	2
Inform jam ahead	Navigation	System	1	3	1
Show speed limits	Navigation	System	2	3	2
Guidance instructions	Navigation	System	3	3	2
Write/send an e-mail	Online Applications	User	2	2	3
Search information in the Internet	Online Applications	User	2	2	3
Create a reminder	Online Applications	User	1	1	3
Read social network posts (text)	Online Applications	User	2	2	3
Shows weather forecast in destination	Online Applications	System	1	1	2
Make a call	Phone	User	1	2	2
Send a message (text writing)	Phone	User	1	1	3
Read a message (text input)	Phone	User	2	3	3
Add a contact	Phone	User	2	1	3
Receive a call	Phone	System	3	3	2
Receive a message (notification)	Phone	System	2	3	2
Add preset to favourite	Radio	User	2	2	2
Change to next station	Radio	User	3	3	2
Set inside temperature	Vehicle Controls	User	3	2	2
Adjust external lights	Vehicle Controls	User	3	3	1
Show actual speed	Vehicle Information	System	3	3	1
Show remaining fuel	Vehicle Information	System	3	2	1
Show current fuel consumption	Vehicle Information	System	2	3	2
Show oil pressure	Vehicle Information	System	2	1	1
Show next maintenance date	Vehicle Information	System	1	1	2

Supposing a very high complexity level where the driver should focus more attention on primary driving tasks, several functionalities can be adapted. For this purpose, a strategy could be adapt all those functionalities with values of “Relevance to user” lower than 2 and “Interaction complexity” higher than 2. As a result, just the following functionalities will be available without any adaptation: Inform low pressure in tires, Alarm about fuel level, Listen to music, Inform jam ahead, Adjust external lights and Show actual speed.

Design proposals for each context implemented is summarized below. In each context a set of changes, based on previously exposed design principles, are applied.

Main Menu

This screen presents a list of available applications. The baseline model distributes these application in a carousel arrangement where sliding right or left the carousel starts a rotation animation. As a new proposal design, applications are ordered in a more simple structure, a grid. Figure 5.2 presents the difference between these two representation format. In the adaptive system, applications are presented in a grid format except for very-low complexity cases (e.g, vehicle stopped).

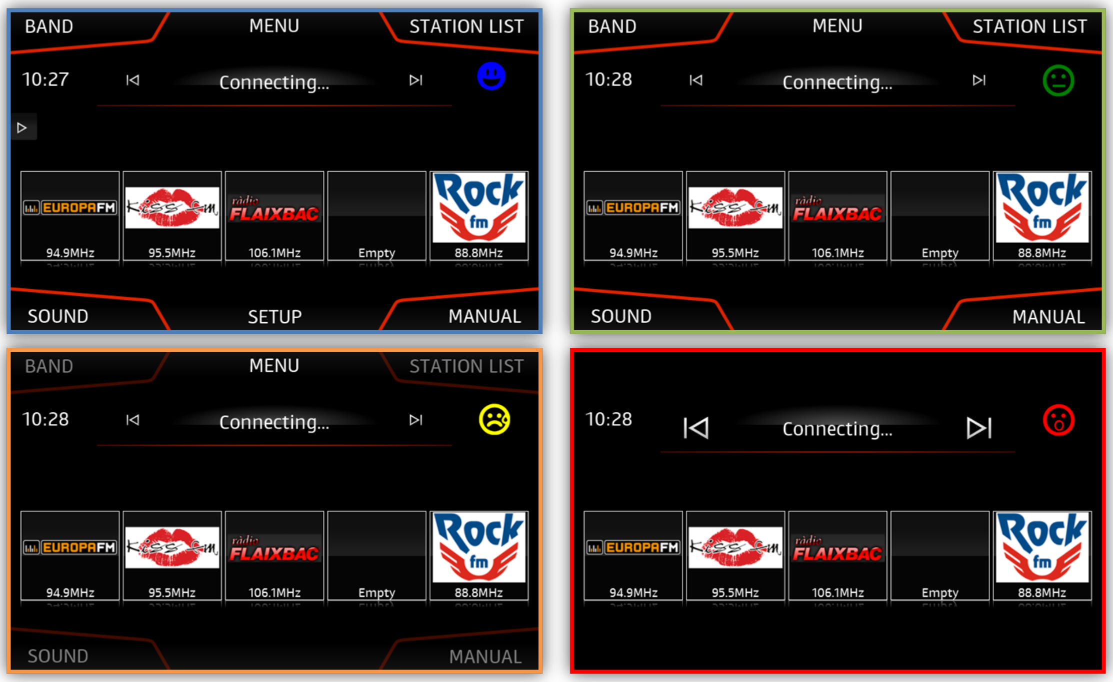
Figure 5.2: Adaptive Main Menu (Very Low (top-left), Low (top-right), High (bottom-left), Very High (bottom-right)).

Additionally, in some complexity levels certain applications can be restricted and not shown in the list. This is the case of the Gallery Application; it can be observed that the icon associated to it (Camera) is removed from the grid in the change from Very Low to Low. The transition from a carousel to a grid distribution is complex and therefore breaks a smooth transition which is highlighted in the Representation of information principle. This strong change was incorporated with the goal of identifying user appreciations about this kind of alterations.

Radio

This contexts allows the user to tune radio stations, change band (AM,FM and DAB), and save favourite presets in a list. In the baseline design (low driving complexity) all functionalities are available, in second complexity level a panel that indicates secondary navigation information is removed (deployable using arrow in left side of the panel). In the third complexity level, actions associated to finding of station in lists and configuration of favourite presets is restricted (corner buttons). Finally, in the last complexity level it is only allowed to change between predefined presets or next-previous stations. Additionally, the size of the buttons is increased. Figure 5.3 shows the changes performed during different driving complexity levels.

Figure 5.3: Adaptive Radio Context (Very Low (top-left), Low (top-right), High (bottom-left), Very High (bottom-right)).

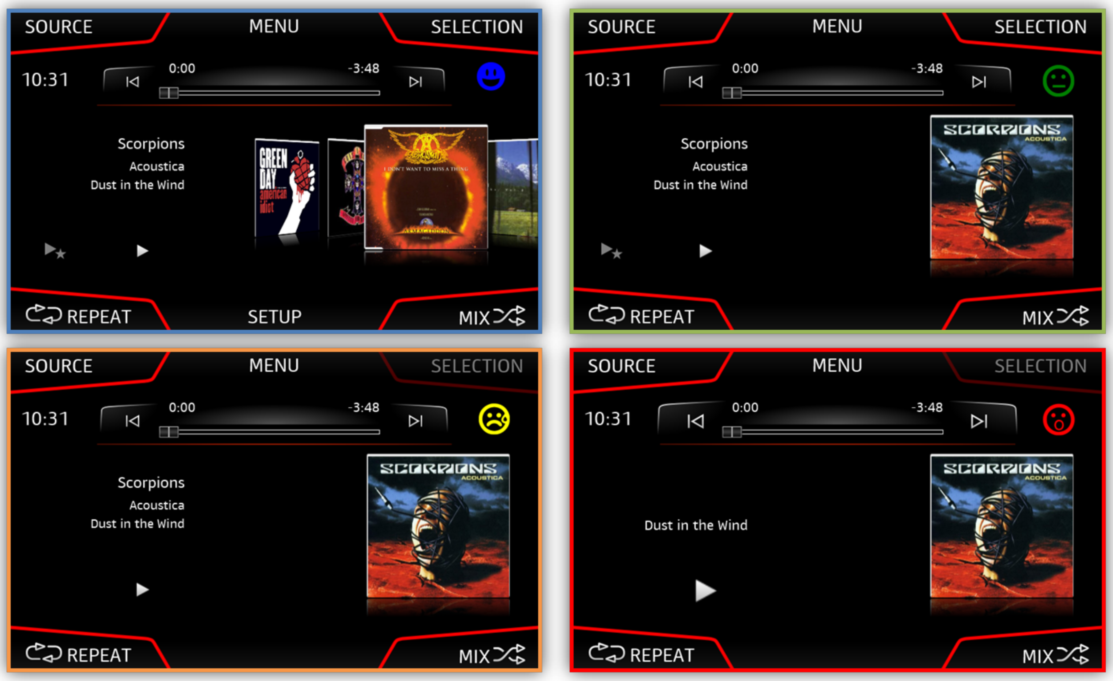


The third (High driving complexity level) and fourth (Very High driving complexity level) designs are similar. In both cases functionalities accessed trough corner buttons are removed, but the approach followed to do it is different. In the first case buttons are disabled and shown with a different opacity value while in the second one, buttons are hidden to provide a more simplistic screen. Different proposals would allow the evaluation of user feedbacks about the benefits and drawbacks of each design.

Media

Media context is used for playing music among available albums. The baseline model (very low driving complexity) allows to select among different audio sources, albums and execute play and pause actions. When the complexity increases some functionalities are restricted (disabling top-right and hidden bottom-centre buttons) which allows the user to search songs and change audio settings. In the next level, the sizes of “next”, “previous” and “play-pause” buttons are increased and the deployable album selection is collapsed (fold-up), finally some details about the album are hidden to simplify the screen. Figure 5.4 shows the flow of changes from the baseline design to the highest driving complexity level.

Figure 5.4: Adaptive Media Context (Very Low (top-left), Low (top-right), High (bottom-left), Very High (bottom-right)).

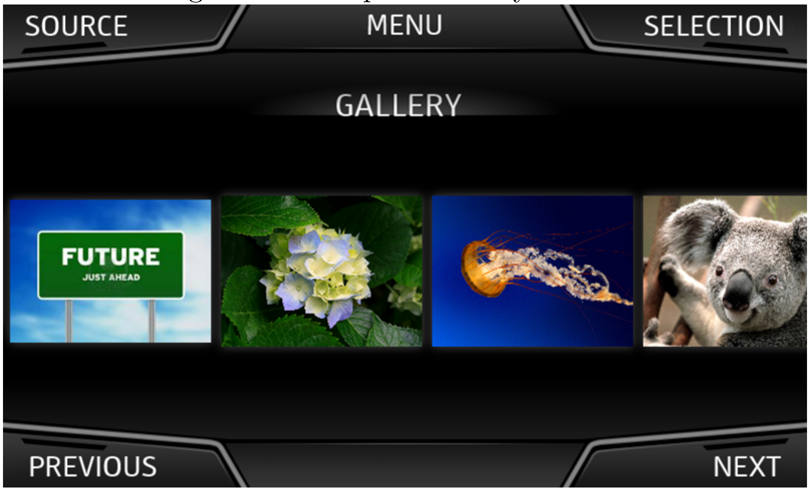


The right corner button gives access to an screen where a list of available songs is presented. The list can be ordered and filtered by song name, album, genre or author. If driving complexity level increases when the user is in this screen, the functionality is not intermediately restricted, but instead it is wait until the user complete the action and return to the Media main context. This approach is followed to avoid user experience problems, so that if users are already engaged in a functionality it is not blocked until the desired task is completed (Adequate timing principle).

Gallery

This application presents a viewer of images. The context design is very simple since it only contains two buttons for changing the image to the previous or next one. The design is not modified depending on the driving complexity due to its simplicity, but instead the application is completely removed from the main menu when the complexity reaches to High. If the user is interacting with the application at the moment of the change, the application is kept open until the user closes it. This approach follows the Adequate timing principle in order to maintain a proper level of user experience. Figure 5.5 shows the Gallery application integrated in the system.

Figure 5.5: Adaptive Gallery Context.



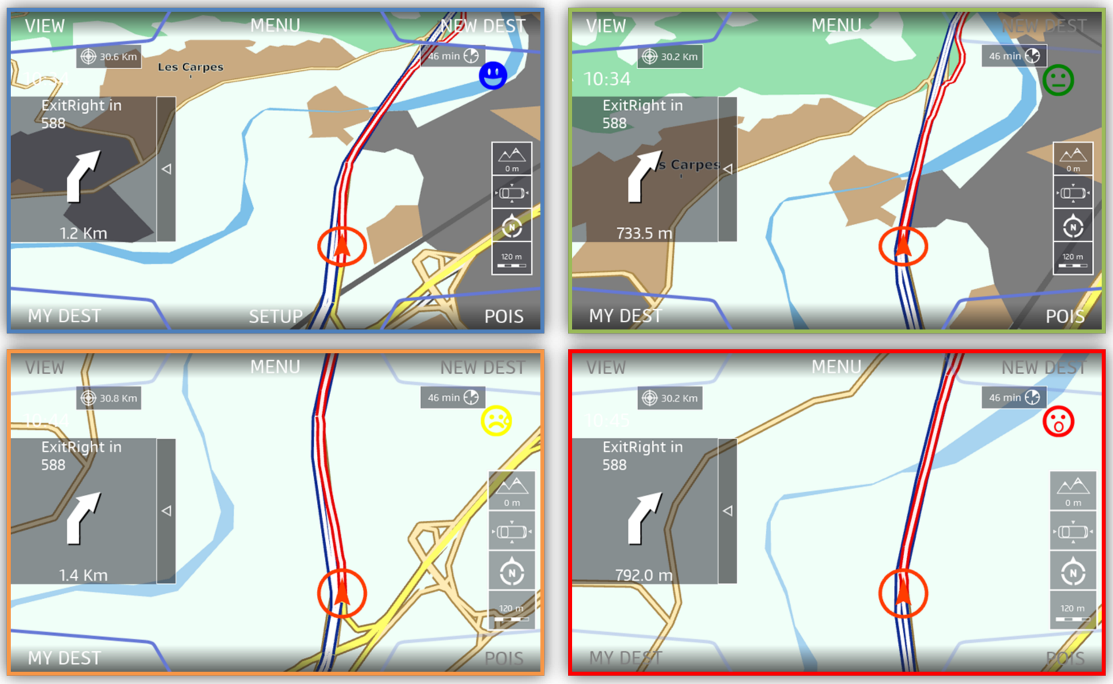
Navigation

The navigation context is very dynamic and complex since it constantly renders a map that sometimes can contain numerous graphic elements. The baseline design contains a button in each corner which allow: setting a new destination, changing the view, selecting pre-configured destinations, and selecting POIs. Two additional panels are presented in each side of the screen showing: (1) Navigation manoeuvres and (2) Configuration of view parameters as zoom and orientation. In the centre of the screen is presented a map which can be centred in the current vehicle position or any other location depending on user interaction.

Figure 5.6 shows the design changes applied for each driving complexity. When the complexity increases to the second level, the button "New Destination" is blocked since for its use, typing in a keyboard is mandatory. For the third level (High), two other buttons in the corner are blocked: "POIs" which is useful for finding places by category, and "View" that allows changing the camera angle. Finally, in the last complexity level, all buttons are blocked (disabled) and the size of the side panels is increased.

A map contains numerous elements and areas representing different information. While driving, not all this information is mandatory for understanding the guidance and therefore maps' appearance is also modified depending on the complexity. In higher complexity levels (above or equal to high driving complexity), some areas, labels and roads are set invisible as: industrial areas, park areas, building areas, town labels with population lower than 50.000 inhabitants, tertiary roads, pedestrian roads and POIs.

Figure 5.6: Adaptive Navigation Context (Very Low (top-left), Low (top-right), High (bottom-left), Very High (bottom-right)).



5.2.3 Test procedure

Based on previous studies, a set of HMI design principles for adaptive interfaces were defined and then applied for modifying and existing HMI design that could be embedded into a Head Unit. The implementation was carried out in a real vehicle, since the purpose of this study is to evaluate the performance and user acceptance in real driving scenarios. The adaptive system was created using as baseline a commercial infotainment system belonging to the automotive manufacturer SEAT (Media System Plus 2015 Interface). This approach enables us to evaluate the real effect of the dynamic nature of the system in a commercially available developed interface.

HMI changes were determined by an indicator that estimates the driving complexity based on 5 variables. These variables exert a significant influence on driving complexity: road type, vehicle speed, wheel angle, traffic, and day/night state. Neither environmental conditions nor profile information was used to predict driving complexity, since all the trips were conducted in similar conditions (partly sunny day on a dry road), and the system was built to be as generic as possible in terms of subjective appreciation of diverse profiles.

The adaptive design was implemented over the most common application context used while driving: Media (location and reproduction of storage songs); Radio (tuning a radio station and saving stations as favourites); Navigation (Guidance-related task, finding location and presentation of guidance manoeuvres); Vehicle Status (Information about vehicle variables and current trip), and finally, an Picture Viewer (browsing of pictures), a more hazardous application to use when driving.

Once the system had been designed and embedded in the vehicle, an acceptance test was executed by selecting 15 participants (10 male, 5 female) who were asked to evaluate the system after using it on a previously defined route. According to Nielsen (Nielsen and Landauer, 1993), 5 participants are sufficient to detect 75% of usability problems, but since this test is also designed to evaluate user acceptance, the number of selected participants was increased to 15. Participants were selected in order to obtain a homogeneous sample in terms of age, driving experience (more than 30,000 km driven), driving periodicity and how much they liked the technology, as shown in Table 5.3. Participation in the test was voluntary and informed written consent was taken from every participant.

Table 5.3: Participants' Profile.

Participants	Gender	Age	Driving Frequency	Driving Style	Appreciation by technology
15	10 males, 5 females	28 ± 4	12 daily, 3 half-month	11 average, 3 sportive	High (15/15)

The route driven in the test consisted of both complex and easy driving segments, including different road types such as motorways, secondary roads and tertiary roads with curves. All the participants characterized the route with the same perceived complexity and encountered similar levels of traffic. The route had a total length of 38 km and was travelled in an average time of 70 minutes. Figure 5.7 shows the route followed and the average complexity predicted by the estimator in each segment; colours represents the complexity: Very Low (blue), Low (Green), High (Yellow), Red (Very High).

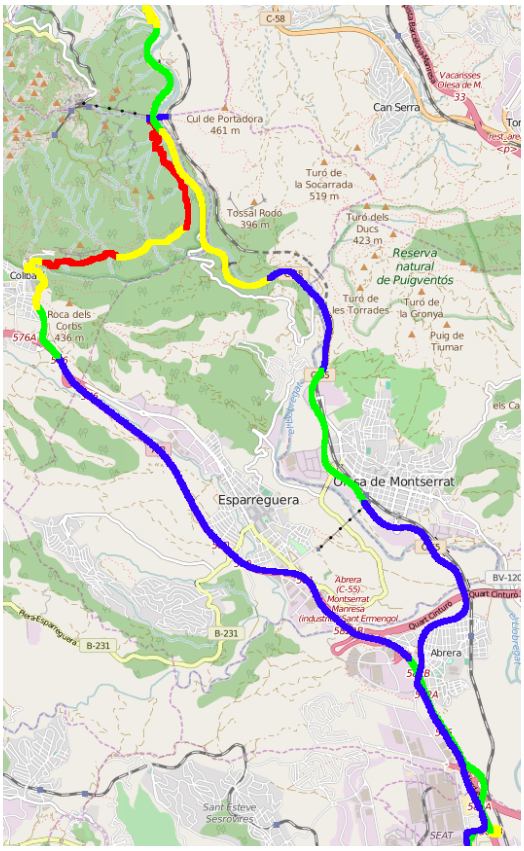


Figure 5.7: Road used for the test and complexity segments.

During the trip, the observer asked the participant to execute some actions on the HMI. These interactions were requested because in common driving scenarios the user may be unlikely to perform many actions while complexity was high. All participants were able to observe all the changes in contexts as well as the adaptive nature of the system.

On completion of the trip, participants were asked to fill out a questionnaire in order to validate three main concepts: evaluation of route driven in order to establish a consistency of route scenario perception; evaluation of the system as regards road safety benefits, and overall acceptance of this kind of system to be used while driving.

5.3 Results

All the participants characterized the route with the same perceived complexity, which enabled us to ensure a homogeneous driving scenario in the tests. As regards acceptance of the system, Table 5.4 shows some aspects of what participants thought when evaluating the use of this type of system; the bottom row may be regarded as a summary of the results reflecting how much the participants liked the system. In general, this constitutes a high level of acceptance, even though some functionalities were highly modified in certain scenarios and affected the common expectations of users when interacting with the interface. One may also see how most of the changes were detected by drivers and had no negative impact on user experience.

Table 5.4: Evaluation of acceptance of the system.

ISSUE REGARDING ACCEPTANCE	AVERAGE	ST.DEV.	MIN	MAX
Detection of changes in the System (From 1 (None) to 4 (The majority))	3,40	0,74	2,00	4,00
Interference of adaptive system on expected use (From 1 (None) to 4 (Much))	2,47	1,36	1,00	4,00
Inconvenience arising from changes (From 1 (None) to 4 (Much))	1,53	0,74	1,00	3,00
Getting use to this system after time (From 1 (None) to 4 (Much))	3,33	0,90	1,00	4,00
Like having this system in his vehicle (From 1 (None) to 4 (Much))	3,40	0,91	1,00	4,00

A parameter very important to consider in this kind of test is the drivers' adaptation to the system, summarized in the third row, "Getting use to this system after time". This value is high, even when the system had not been previously shown to participants and therefore they were not aware about the system behaviour. A frequent use of this kind of systems is expected to increase drivers SA level, this will allow users to recognize in advance the changes about to happen.

The lowest value in the table was found for the "Inconvenience arising from changes"; this question was oriented to determine how the changes affect the desirable use of the system. The value indicates that some of the changes may have been too intrusive and have affected the real functionality of the application. Participants were asked in this regards and they indicated that some restrictions in Radio or Media limited the overall purpose of the application. This issue must be careful studied, since the system must not only assure road safety but also guarantee a proper level of user experience. In consequence, future designs should study how to offer the access to more functionalities in higher driving complexity levels without affecting driver workload, e.g., The button used for changing the BAND in Radio context is kept for every scenario.

Table 5.5 shows the results of the questionnaire on road safety. These results show that participants tend to have a high appreciation that this type of system would help prevent accidents and increase road safety. Participants were also asked about the inclusion of new functionalities and applications in low-complexity scenarios, rather than limiting their interaction completely while driving, showing a positive feedbacks towards these additions.

Table 5.5: Perceived Road Safety.

ISSUE REGARDING SAFETY	AVERAGE	ST.DEV.	MIN	MAX
Perceived impact of the system on road safety (From 1 (Negative) to 4 (Positive))	3,73	0,46	3,00	4,00
Opinion on integration of new functionalities in low-demand scenarios (From 1 (Non-Adequate) to 4 (Adequate))	3,53	0,64	2,00	4,00
Opinion on restriction of functionalities in high-demand scenarios (From 1 (Non-Adequate) to 4 (Adequate))	3,73	0,59	2,00	4,00

5.4 Discussion of Results

Test results show a positive feedback of participant towards the inclusion of adaptive interfaces in vehicles. The design principles applied have shown to provide a good user experience as well as an increase in perceived road safety. These principles must be clearly aimed at an understanding of user preferences in terms of the functionalities that are most frequently used. Furthermore, one of the main premises of such adaptation is the introduction of HMI changes without having a high impact on the ultimate purpose of the application.

Another relevant opinion expressed by participants concerned the need to ensure a smooth transition process of adaptation when the road complexity changes. Moreover, changes must be stable over time and only be activated if the driving complexity is the same over a certain period of time; this would obviate highly dynamic interfaces that could affect user experience. One aim of the design is that regular users should not be aware of most of the restrictions, since in complex scenarios should not be common a frequent use of these limited functionalities.

The importance of providing an accessible indicator showing the current driving complexity was also detected. This enables drivers to understand and remember what restrictions are being adapted. In the test it was found that, although the participants did not know in advance what the HMI changes were, they did not feel confused when the restriction was activated. The proposal for a future test is to evaluate the impact of the system when it is used on a regular basis in drivers' own vehicles, where driving behaviour is deemed to be more natural.

With regard to the adaptation of more commonly used features, positive feedbacks were received. All the participants thought that the increase in button size would help to reduce the glance time when performing a very common action such as changing tracks or radio stations. Also, most participants preferred that unavailable or restricted functionalities be completely hidden rather changing contrast, since the opacity of a feature is not always understood as an unavailable feature. Nevertheless, this is a controversial point and mixed opinions are expressed about the benefits of each alternative.

5.5 Summary

Due to the increasing amount of functionalities and services that are being added in driving scenarios, it is necessary to create a system capable of managing the flow of information. This section introduce a set of design principles that can be used for developing adaptive interfaces based on the current driving complexity. Most of these principles are based in previous recommendations and design guidelines for designing IVIS. Proposed principles were classified in 5 main concepts: consistency, modality, adequate timing, situation awareness an representation of information.

Under this approach, an Adaptive HMI (A-HMI) was developed using as reference the HMI of a commercial Head Unit created by SEAT. The interface was modified in a way that several changes were applied restricting some functionalities or making some alteration in the design according to the current driving complexity. The system, once integrated in a vehicle, was validated performing acceptance and usability test in real driving scenarios.

As result, the system has shown to perform well in real driving scenarios and positive feedbacks were received from participants endorsing the benefits of integrating this kind of system as regards driving experience and road safety. This test enabled to define a strategy to be followed that is based on some of the design principles evaluated herein. The main benefits of the system in terms of road safety will become evident once drivers are accustomed to the behaviour of the system and understand that in certain scenarios complex functionalities may be restricted.

An adaptive system can contribute in different manner to road safety, the first is trough the limitation of functionalities that clearly affects the performance of primary driving tasks, the second can be the easing of information presented to the driver according to its need in different scenarios. Thirdly, the driver by itself may adopt lesser aggressive drivings when he wants to perform an action that is restricted as a consequence of a complex scenario. As example, the user wants to search an album, but given that he is travelling at higher speeds the functionality is restricted, he could avoid going at such speeds in order to have a less limited access to these functionalities.

As a future support to this type of system, a series of tests is proposed in order to obtain measurable values for comparing the benefits in distraction times; for example, a response-time test employing mechanisms such as the Peripheral Detection Task (PDT) or an unawareness detection-time by means of Eye Tracking monitoring. Other considerations that must be taken into account in the strategy for a commercial implementation of the system are: inclusion of the driver profile as a metric for personalizing the interface as regards user preferences, and control of the functionalities that should be changed to a greater or lesser extent. A further relevant point is the detection of a passenger as copilot, since in general complicated tasks would be carried out by the passenger, although it is possible that such copilots may not know how to interact with the system or may have some impairment, and in this case the strategy may not fulfil the aims of the system for which it is designed.

Chapter 6

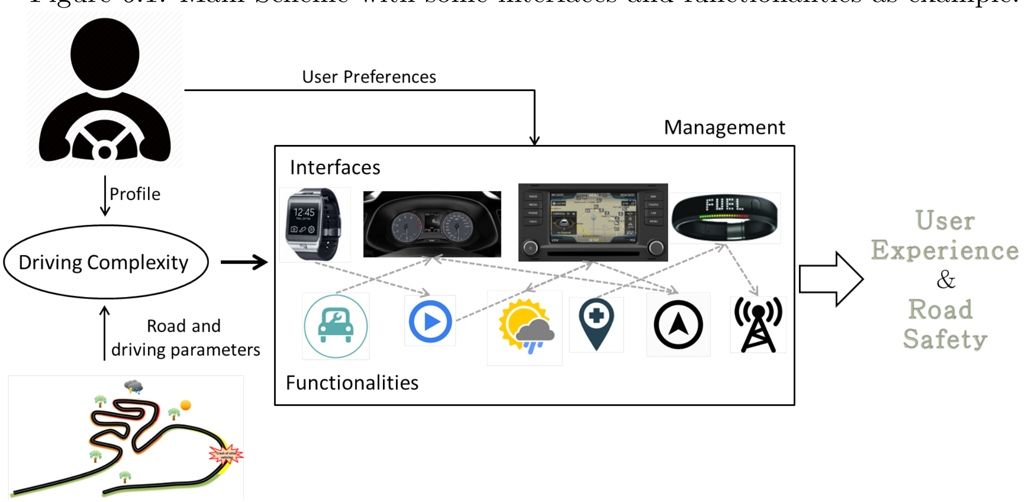
Multiple Interfaces while driving

Current vehicles are beginning to integrate every type of interfaces for showing information and interacting with applications. In some cases, these integrations have clear benefits in regards of road safety, but in others their intention is mainly to improve user experience. The benefit of using an interface while driving is a result of its capabilities for simplifying the interaction and reducing the execution time of a task, and also by the functionality incorporated, e.g., a voice command may be very useful for initiating a call but annoying for simple tasks such as changing volume or jumping to the next song.

In order to achieve a good balance between road safety and user experience when multiple interfaces are available, a proper distribution of information and ways of interacting with functionalities must be guaranteed. This chapter provides an upgraded set of design principles for cases in which multiple interfaces are available while driving.

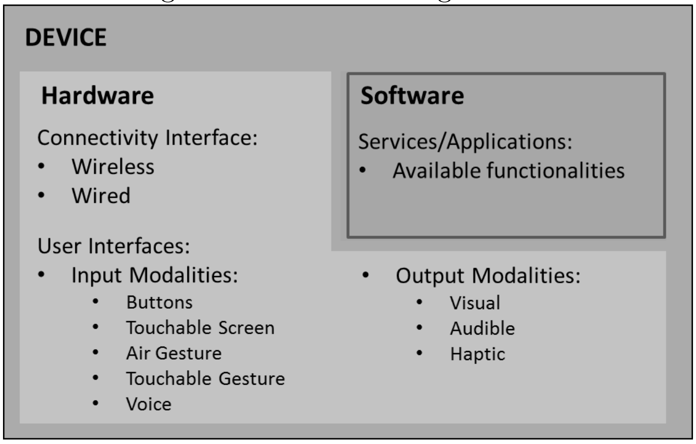
Figure 6.1 shows a summary of an hypothetical adaptive system composed by several user interfaces. In this scheme, two main metrics are exposed as reference for managing the representation of information and interaction with functionalities in several interfaces: User Preferences and Driving Complexity. Each interface (e.g. Head Unit, Kombi, smartwatch, ...) may offer different applications, in some cases redundant; therefore the final arrangement of distribution of information must be arranged each time a new interface is added to the system.

Figure 6.1: Main Scheme with some interfaces and functionalities as example.



The initial step to create the multi-interface system consists in the identification and classification of devices in terms of: functionalities offered, output interface (how the information can be transmitted to the user e.g. visual, audible, haptic) and input interface (how the user can interact with the device, e.g. voice, gesture, touch). Figure 6.2 shows a proposal for classifying device in terms of offered functionalities (mainly software layer) and connectivity and user interfaces (hardware layer). The figure shows some examples of possible user interfaces that can be used.

Figure 6.2: Device Categorization.



Once all the devices and functionalities that will be incorporated into the system are identified, a characterization table summarizing all these data will allow to determine what are the different options for representing each information and interacting with each functionality. Here, each functionally-interface relation must be carefully analysed in terms of three metrics: relevance to the user (how much the impossibility of accessing a functionality blocks an interaction), safety impact of accessing to it while driving (road safety impact) and user preference (given by each user).

6.1 Extension of Design Principles for Multiple Devices

An extension to the principles proposed in Section 5.1 taking into account the addition of multiple interfaces is presented below. This extension is based on Schneiderman's Eight Golden Rules (Shneiderman (2010)), Jakob Nielsen's Heuristic Usability Principles (Nielsen (1993)) and IVIS Guideline Recommendations (ESoP (Commission of the European Communities (2008)) and NHTSA (NHTSA (2012))).

- Consistency: Information should be provided in a standardized manner, not only for the an interface but among all connected devices. This consistency should be maintained when it respects to syntaxes of sentences, words, icons, colours, style, sounds and vibrations. The consistency is also relate to how the adaptation is made in every device, each one must offer a similar transition of adaptation depending on driving complexity.
- Modality:
  - Regarding Wicken's model, interfaces should not present information trough same modalities at the same time. For example, an incoming phone call ring

can be delayed a few seconds if a warning message is emitted at that moment trough an auditory channel. Another approach could be the notification of the incoming call using another channel, for example visual.

  - Information should be clearly represented selecting the more proper interface according to the current driving complexity and user preferences. The same approach should be followed in regards of interactions; if a functionality is too complex to be managed using a given interface, a restriction mechanism should avoid its use.
  - The driver should be able to configure certain interfaces' properties such as the loudness of auditory information, the brightness and contrast of visual interface and the vibration intensity of tactile interface. A minimum threshold value must be defined for high priority notifications. Similarly, the system must automatically avoid that low priority messages mask high priority notifications or warnings.
  - Warning and emergency alters must be clearly identified by the user. In order to avoid confusions redundant visual information is proposed to be added. This information must be very simple and clear to avoid an overload of information.

- Adequate timing: For every interface, information should be transmitted following a coherent priority scheme based on road safety impact and user preferences. Also, the short-term memory should be reduced (avoiding that the user has to remember information from one screen to another) and drivers should be able to cancel or reverse an action with a single interaction.
- Situation Awareness:
  - The user should be capable of predicting the overall behaviour of available interfaces and figure out where the information needed is presented. Also, he should be able of understanding why a given interface is not showing some information and why some restrictions have been raised.
  - The system should always provide informative feedback about the system status for each device indicating processing states, operation results, restrictions, errors and alternatives to operations. In case that an interface is restricting certain functionalities, a comprehensible informative status should be presented. This indicator is proposed to be standard among all devices.
  - A proper approach to follow when the user wants to execute an action and it is restricted, could be to offer him alternatives. As an example, if the driver wants to set a new destination interacting with the touchable screen, the system can start a voice recognizer system asking the driver to tell the address instead of needing the screen. This mechanism avoids the driver to deviate the sight, reducing the impact on road safety.
- Representation of information: This principle should be maintained for every interface conforming the overall system. Its main goal is to permit the user detect and comprehend the message as fast as possible while maintaining a good level of user experience. As example, information presented in a HMD may offer a higher user than using a Head Unit, or the vibration of a smartwatch may be less less annoyance than a sound for indicating a notification.

6.2 A Proof of Concept: Head Mounted Display vs Kombi

In order to analyse the advantages and disadvantages of adding new interfaces in a vehicle, two user interface were compared in terms road safety and user experience: a dashboard which is a commonly used interface (baseline) and a Head Mounted Display (HMD). Head Mounted Displays may offer an easier and faster interaction with information, services and applications since the driver does not have to divert much the sight from the road for visualizing information. The integration of a HMD into a vehicle could be useful in driving environments where the visualization of relevant information and the interaction with services could compromise safety. For comparing the benefits, the vehicle Kombi (panel located in the dashboard) is used as reference.

This section is not devoted to explain the technical procedure for connecting different interfaces in a vehicle (connectivity interface). It is assumed that each device can communicate with each other and the information can be shared without restrictions. Moreover, the user interaction over one device can have the effect of initiating an action on another; e.g, a gesture on a smartwatch can turn on the vehicle lights, or an abrupt acceleration can deactivate a head mounted display.

As HMD, a Google Glass device was selected for performing the tests. It was programmed to have certain set of functionalities already available in the vehicle: *"Vehicle Status"* that presents information about vehicle variables such as warnings and overall vehicle state, *"Coaching"* that informs the driver about gear shift recommendations, and *"Climate – Audio Source Setup"* that provides some services such as changing of media source (Radio AM, Radio FM, USB, SD), and climate control. These services can be controlled trough touchable gesture (Google Glass touch-pad) or voice commands.

6.2.1 Material and methods

Usability and distraction tests were performed in order to determine the advantages and drawbacks of interacting with a HMD in driving scenarios. Data used for the usability test comes from participants questionnaire and results were compared in terms of user experience and user acceptance. In case of the distraction test, data was collected from glance patterns using a video camera and driving performance accessing vehicle variables thanks to an instrumented vehicle.

Three functionalities were integrated in the HMD for its evaluation: (1) a driving oriented application called "Coaching", which indicates the current gear and recommended gear in regards of ecological driving, (2) an application called "Vehicle Analysis" which allowed to request information about the vehicle status, and (3) "Media/Climate Control" which allowed to set the temperature inside the vehicle and select the audio source.

Displays impact (HMD and Kombi) were compared in a controlled driving scenario. For this purpose, 12 participants were asked to drive a predefined route following the instructions of recommended gear given by the display (Kombi or HMD). Just one of these two display was available per route (independent tests), given that the purpose was to evaluate the impact of each one when used separately. The information presented in the Kombi was more extensive than the offered in Google Glass; nevertheless the functionality was the same: presenting the actual and recommended gear when necessary for indicating a change.

Participants received an introductory course for explaining them how to use a Google Glass unit, which application they were going to use, the purpose of the test and how they will be evaluated. After each test, participants had to fulfil a set of questions related to the workload appreciated; the metrics were:

- Global Demand: Overall demand associated to the constant visualization of the recommended gear indicator. It is related to the additional load caused by thinking and deciding what to do in relation to the additional task.
- Visual Demand: It is related to the additional difficulty caused for the constant change of focus; this task is associated to periodic glances of the indicator.
- Stress: Related to the level of fatigue, insecurity, discouragement or nuisance executing the task.
- Temporal Demand: Restriction of time for continuous monitoring of the displays, feeling that the task was affecting responses time.
- Interference: Related to the interruption and affection of a normal driving behaviour.
- Comparison with regular driving: Comparison with a regular and daily driving about how a user would evaluate himself in relation to the driving performance during the test.

Questionnaires are added in the Appendix Section (C).

System

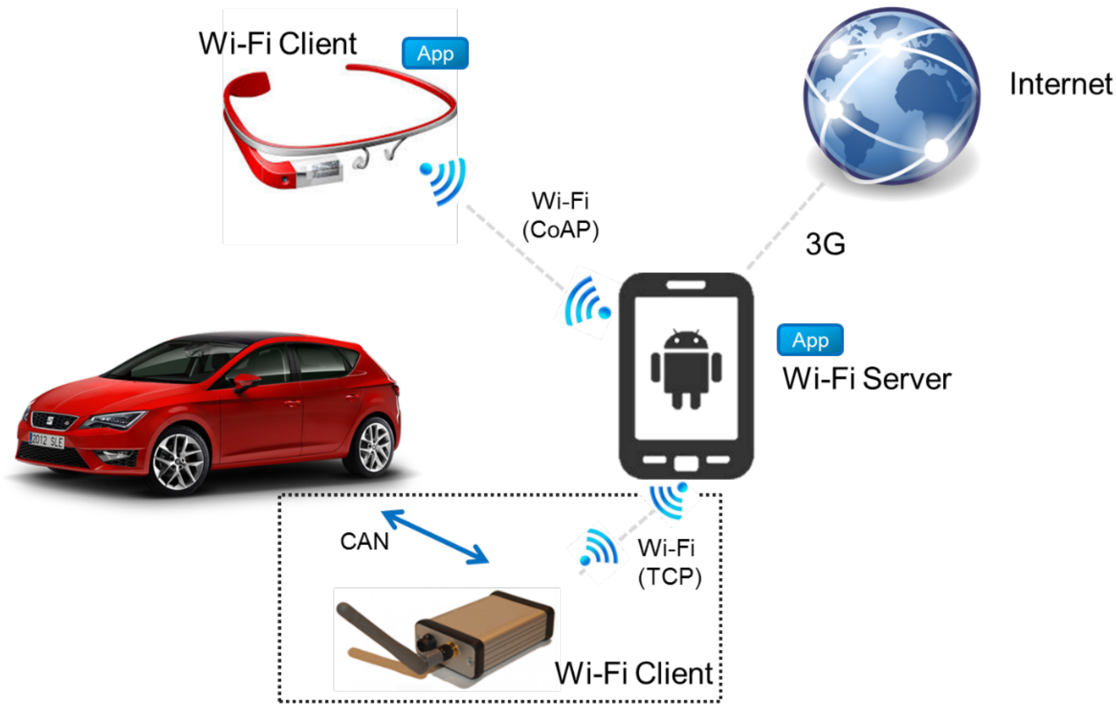
The architecture needed for integrating a HMD in the vehicle depends on the communication interfaces provided by the HMD. As a Google Glass device integrates a Wi-Fi and a Bluetooth interface, it was decided to use the Wi-Fi interface for exchanging information. The Bluetooth interface was kept for getting internet access from a smartphone. Google Glass Specifications are presented below:

- Hardware:
  - CPU: OMAP 4430 SoC, dual-core.
  - RAM: 1GB (686MB available for developers)
  - Storage: 16GB
  - Battery: 570mAh. (Autonomy: 1day)
  - Screen Size: 640X360 (translucent display)
  - Connectivity: Wi-Fi 802,11b/g, Bluetooth.
  - Components: Camera: (5Mpx, 720p), microphone, accelerometer, gyroscope, magnetometer, proximity and light sensor.
  - No embedded GPS.
- Software:
  - Android OS (v4.04, API 15 (Ice Cream Sandwich))
  - For developing purposes Google facilitates a special development kit (Glass Development Kit (GDK)).

Figure 6.3 shows the architecture implemented. The smartphone is used as communication bridge between the Google Glass unit and vehicle variables. The smartphone access variables communicating with a Wi-Fi module which physically access the Bus CAN and creates a TCP socket for sharing information.

In the smartphone a background service creates a Representational State Transfer (REST) server based on a protocol called Constrained Application Protocol (CoAP). This server makes available a set of variables that can be accessed by CoAP clients. Google Glass act as a CoAP client and subscribe to resource for reading updates, it can also send PUT commands to request an update of a variable in the vehicle, e.g, the temperature of the air conditioning. Most part of the applications were implemented in the Google Glass device and for developing the CoAP communication service it was used an open source Java library called Californium Framework (Provided by Eclipse Org).(Kovatsch et al. (2014)).

Figure 6.3: Google Glass system architecture.



REST and CoAP

REST is a model that defines with a set of principles how web standards, such as HTTP and URIs, are supposed to be used. The model follows a Client-Server stateless scheme where each query contains all the information necessary to understand the request. In this model, item of interests are called resources and are associated to unique IDs called URIs. The use of global namespaces to identify resources facilitates a quick understanding of the scheme (Tilkov (2007)).

One of the main principles of REST is the creation of an adequate relation between resources, in this regards resources can be categorized and accessed easily from any service or third party application. If URIs are well constructed, it is easy to understand the meaning of resources, its origin and characteristics. The following example shows the URIs for a system that contains a set of variables for different categories:

- http://system.com/sensors/12
- http://system.com/actuators/mechanicals/3

The access to resources is done through standard methods (GET, POST, PUT and DELETE). Another useful method is OBSERVE, which follows a subscription scheme and updates are automatically received using a predefined timer or only when the resource changes. Thanks to this later approach, unnecessary requests of information are avoided and the bandwidth overload is reduced.

CoAP is an application protocol specially designed for constrained networks in terms of bandwidth or when devices have limited energy capacities. For this reason, CoAP has been adopted as one of the preferred Internet of Things protocols. It was created by the CoRE (Contrained Resource Environment) IETF group as an improvement of other existing REST protocols that did not take in account bandwidth or energy constrains. This protocol was registered in 2014 as the RFC 7252 (Shelby et al. (2013)).

Some of the main features of CoAP are:

- Asynchronous messages exchange protocol.
- Lower header overhead and packet size in relation to other protocols as HTTP.
- Is interoperable with HTTP and the RESTful Web through proxies.
- Is datagram based; usually on UDP but it is possible to implement it also over TCP.
- Uses a Client-Server scheme where requests and responses messages can be marked as non-confirmable (NON) or confirmable (CON). Confirmable messages must be acknowledged by the receiver with and ACK packet.
- It allows to implement subscription mechanisms for observing changes of resources values; this is achieved constructing relations of notifications that can be cancelled any time.
- Discovery mechanisms to find available resources are also available.
- The security layer of the protocol is constructed using Datagram Transport Layer Security (DTLS), RFC6347.

CoAP has been implemented in different environments and it has support libraries in several programming languages as C, C++, .NET, Java and Pyhton. In the case of Java, some support libraries are nCoAP, jCoAP, Californium. Among all, the most complete, updated and tested library was found to be Californium, which is now managed by the Eclipse Organization.

Environment

All participants drive the same route with similar environmental conditions, partly sunny day. As the Google Glass display was sometimes difficult to watch under bright conditions, a sun glass adapter was incorporated and the car sun protector was drop down. Each participan drove in average a total of 30 minutes (15 minutes using each kind of display). The route was not complex and traffic density was very limited to avoid road safety problems. Figure 6.4 shows the route defined during the test.

Figure 6.4: Road Map used in coaching test (HMD).

Participants

Twelve participants (7 males, 5 females) were selected to execute the usability/acceptance and distraction tests. According to Nielsen Nielsen and Landauer (1993), 5 participants are sufficient to detect 75% of usability problems. A higher number of participants would be only required if a profile patterns analysis is needed.

Participants' profile data is presented in Table 6.1, where is observed that 58% of the participants were men and 42% women. In relation to the age, approximately one third of participants were younger than 35 years old, one third was between 35 and 45, and one third were older than 45 years old.

Regarding the number of kilometres driven per year and the frequency of vehicle use, the greater portion of participants have driven more than 10.000 Kms per year and have used the vehicle more than one day per week. This indicates that participants can be seen as habitual drivers and its driving experience should not interfere in the results obtained. It was also taken into account the use of lenses, from the sample, half of the participant uses lenses.

Table 6.1: Participants' Profile

Participant	Gender	Age	Kms driven per year 1: <10.000 2: 10.000 -25.0000 3: >25.000	Vehicle use T: Work M: Mobility P: Pleasure	Frequency of vehicle use	Use of lenses
1	M	47	3	T,M,P	Every day	No
2	F	36	2	T	Every day	No
3	F	23	3	M	Every day	No
4	M	44	2	M	Every day	No
5	M	26	3	P	Several days per week	Yes
6	M	48	3	T,M,P	Every day	Yes
7	M	32	2	M	Less than once per week	Yes
8	F	39	1	M	Less than once per week	Yes
9	M	55	3	T,M,P	Every day	No
10	M	24	3	T,P	Every day	Yes
11	F	37	1	T	Several days per week	Yes
12	F	26	1	M	Every day	No

6.2.2 Results

Results are presented in terms of the following metrics: subjective workload perception, glance patterns and driving performance metrics.

Subjective workload perception:

A comparison between the perceived demand using the Kombi (baseline) and Google Glass is presented. Figure 6.5 shows 6 categories where is evaluated the driver perception in terms of: (1) Global Demand associated to the workload caused for interacting with the interface Kombi or Google Glass, (2) Visual Demand which express the additional cost of focusing each screen instead of the road, (3) Temporary Demand linked to time constrains perceived due to the use of each interface, (4) Interference defined as a the degree of affection of driving performance due to the additional task, (5) Stress which indicates the perceived fatigue, insecurity or discouragement and (6) performance comparison with regular driving conditions.

In relation to the Global Demand (Figure 6.6a), participants show a higher demand while using Glass, nevertheless the difference is not too significant to be conclusive. Results obtained were: 2 cases of lower demand using the HMD, 5 cases of equal demand was equal, and in 5 cases in which the demand was higher using a Google Glass Unit. In one of the cases (participant 6), the use of Google Glass was considered much more demanding than Kombi; the observer noticed that this participant transmitted his concern about the road safety impact of using a Google Glass while driving would cause.

In regards of Visual Demand (Figure 6.6b) results are balanced; one third think the visual demand is lower using glass, other third think that is higher when interacting with the HMD, and the final third think the demand associated is the same. The comparison in terms of visual demand is quite important since the interfaces have different screens located in different places.

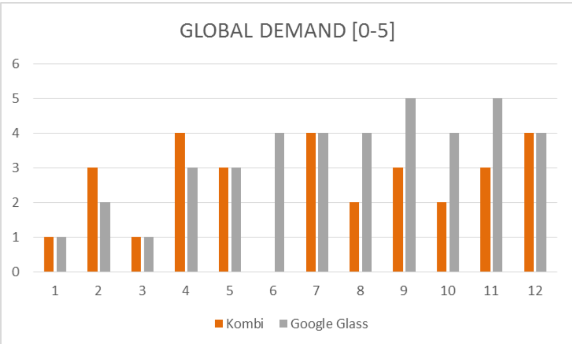
In relation to Temporary Demand (Figure 6.6c), participants did not expose differences between using Kombi or Google Glass. Small difference are observed and and only one participant (identified with the number 8) felt that the temporal demand was much higher with Google Glass. Overall, in 5 cases the demand was lower using Google Glass, in 1 case the demand was equal and in 6 cases the demand was higher using a Google Glass unit.

The interference perceived by participants over the driving tasks is presented in Figure 6.6d. One half of the sample perceived a lower interference when using the HMD and the other half exposed a lower interference when driving and using the Kombi. In most of the cases the difference was smaller (less than 2); participant 8 was the exception showing a higher difference between the two interfaces.

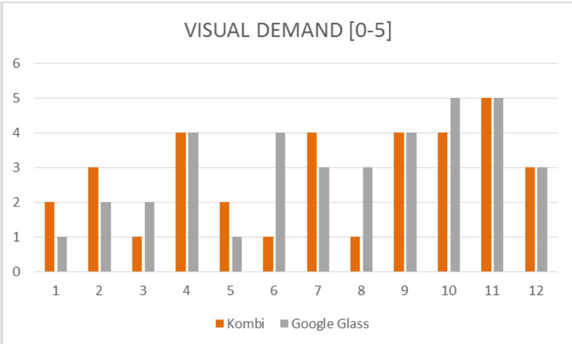
Regarding the Stress (Figure 6.6e); most of participants expressed a higher perception of stress when driving and visualizing the Google Glass display, this could be linked to the fact than they were using this kind of device for their first time, moreover they reported that under high contrast situations was difficult to see the Google Glass display. Results obtained were: in 1 cases the demand was lower using Google Glass, in 5 cases the demand was equal and in 6 cases the demand was lower using the Kombi.

Finally, participants were asked to give their opinion about their driving performance in comparison to their regular driving performance (Figure 6.6f). Using a scale from 0 to 10, where 10 is a good performance and 0 the opposite, 6 participants considered they drove better using Google Glass, 2 considered they drove equally and 4 that the driving was worst using Google Glass.

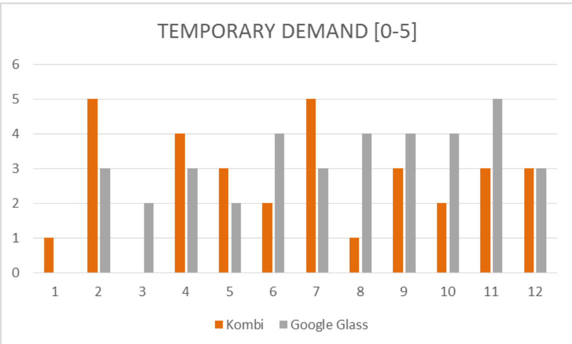
Figure 6.5: Demand Comparison between Kombi and HMD (Subjective Perception).



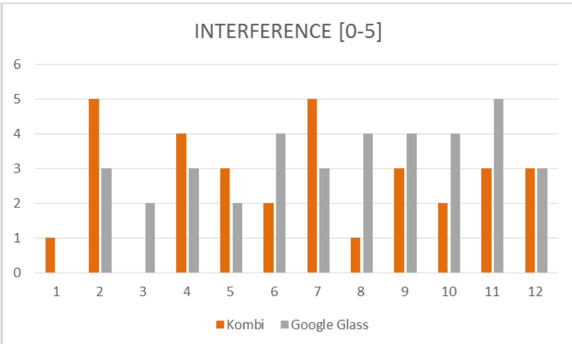
(a) Global Demand



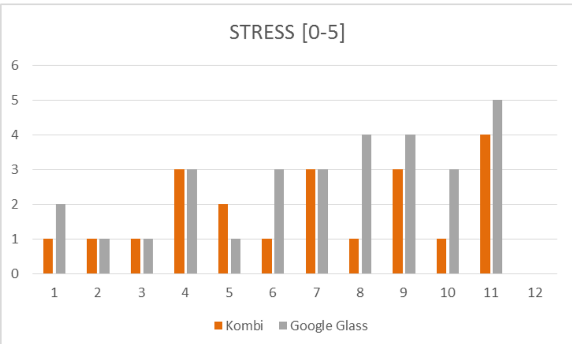
(b) Visual Demand



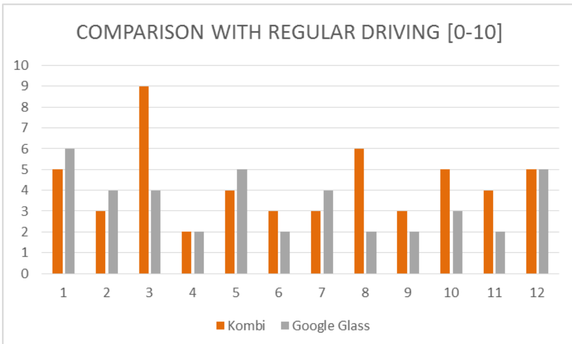
(c) Temporary Demand



(d) Interference



(e) Stress



(f) Comparison with regular driving

Overall, workload perceived by participants were similar when using Kombi and Google Glass. The observer noticed certain level of predisposition in regards to the use of Google Glass while driving, this condition may have affected in some degree the user perception and therefore the results. Some participants were very attracted to this new technology may have reported a better driving experience and on the contrary, participants who refuse the use of this kind of devices while driving may have reported worst results than their real objective perception.

Furthermore, participants were using a Google Glass for their first time, and even when they had time to get used to it, they may have felt uncomfortable. It was commented by several users that after using the device for a time they beginning to feel more comfortable and to have less problems focusing the Google Glass display.

Analysis the relation between profile data and the results, none correlation was found. A larger sample of participants in a future test could offer some additional conclusions in this regards.

Glance patterns:

The frequency and duration of glances toward the display (Kombi or HMD) was monitored in driving scenarios. Data was collected recording participant's faces with a camera and observing their visualization patterns. This test allows have an objective measure about how often they watch the display and for how long.

Table 6.2 shows the average glance time duration and frequency per display and for each participant. The glance duration is defined as the period since the participant looks away to the screen until it returns the focus to the road or any other point other than the screen (could be mirrors).

Table 6.2: Glance Analysis while driving (Comparison HMD and Kombi visualization).

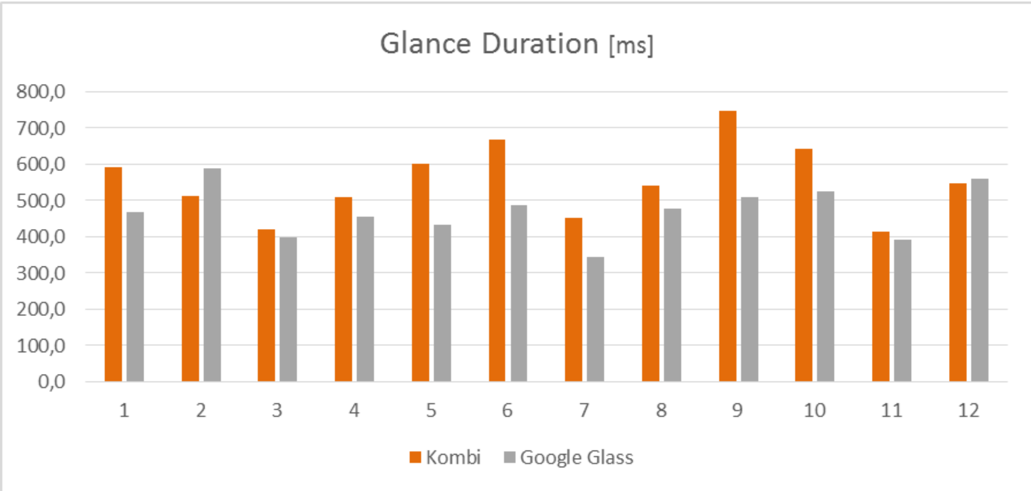
Participant	KOMBI						Google Glass					
	Freq. Glances		Glance Duration				Freq. Glances		Glance Duration			
	Armonic avg.	Prom. (SD).	Armonic avg.	Prom. (SD)	Min	Max	Armonic avg.	Prom. (SD).	Armonic avg.	Prom. (SD)	Min	Max
1	1,7	2,8	590,7	177,7	413,1	768,4	0,9	1,2	467,9	160,4	307,4	628,3
2	1,8	1,6	513,2	110,0	403,2	623,2	1,8	2,5	588,4	134,3	454,1	722,7
3	0,9	4,2	418,8	73,2	345,6	491,9	0,9	1,4	398,2	123,9	274,4	522,1
4	0,9	0,4	509,6	118,8	390,8	628,4	1,0	1,0	455,0	91,9	363,1	547,0
5	0,8	0,7	601,3	146,4	454,9	747,7	0,9	1,1	433,0	138,1	294,9	571,1
6	1,1	0,7	666,2	207,2	459,0	873,4	1,4	1,4	485,4	165,0	320,4	650,4
7	0,7	1,1	451,5	104,5	347,0	556,0	0,8	1,3	343,3	134,0	209,2	477,3
8	1,6	2,7	541,7	179,3	362,5	721,0	1,9	1,9	477,4	234,4	243,1	711,8
9	0,9	2,4	746,7	167,0	579,7	913,7	1,2	1,9	509,4	136,4	373,0	645,8
10	1,0	2,3	643,3	240,0	403,3	883,2	0,7	1,4	524,5	365,3	159,2	889,8
11	1,3	1,4	414,4	83,4	331,1	497,8	1,4	1,7	390,6	49,7	340,9	440,2
12	2,0	2,8	545,6	133,4	412,3	679,0	1,9	1,9	560,3	226,3	334,1	786,6
AVG.	1,2	1,9	**553,6**	145,1	408,5	698,7	1,2	1,6	**469,5**	163,3	306,1	632,8
STD. DEV	0,45	1,14	101,54	50,69	67,59	145,56	0,43	0,45	71,41	81,15	78,56	131,15

The average glance duration is observed to be lower for a Google Glass display than for the Kombi display (see highlighted cells in red). Comparing both times, a Google Glass duration represents the 84% of the Kombi duration. This may be a sign that information is easier to comprehend using a Goggle Glass due to faster focus, or may be a consequence of driver insecurity when diverting the sight from the road.

The frequency of visualizations and time between glances was quite random and also very linked to each participant' behaviour. Based on participants' reports, they tend to look at the displays based on expectations when they believe the vehicle may need a gear shift or subject to their availability to look away from the road according to driving scenarios (curves or more demanding driving situations). For Google Glass and Kombi, the same average of frequency of glances was obtained.

Figure 6.6 shows a graphical representation of the glance times associated to each participant. In the graph is observed that 83% of participants have a lower average glance using Google Glass. As previously exposed, this result may be an indicator that participants have fewer problems viewing the information presented in the Google Glass display.

Figure 6.6: Glance duration comparison per participant.



The result is quite surprising since participants were using a Google Glass for their first time and therefore we could expect certain discomfort. Based on participants' opinions, a training phase was needed for getting use to focus the Google Glass unit.

Driving patterns:

Driving patterns were also compared in order to identify relations between the display used and the driving performance. The elapsing time between a gear change recommendation and the performing of the change was used as metric. Also, the number of times a recommended gear was announced in the display was recorded. These values are proposed to be used for measuring driving performance since participants were explicitly instructed to make the gear change as soon as the notification was announced.

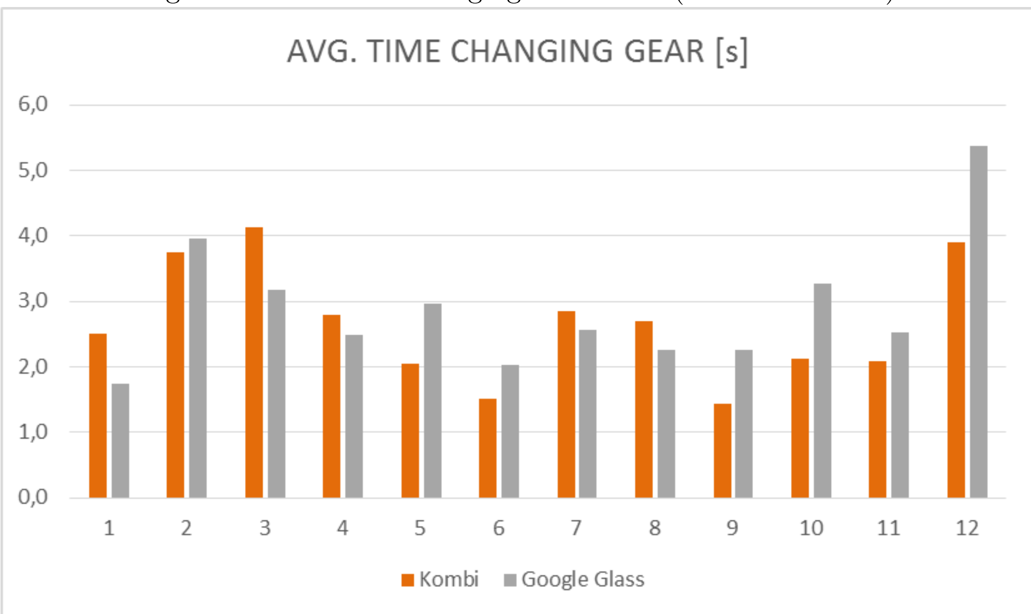
Table 6.3 shows the average time of each participant for changing gear when using the Kombi or the Google Glass as display interface. In average, the amount of time a recommended gear was announce was very similar for Kombi and Google Glass (20 recommendations with a standard deviation of 5), therefore we can assume that the driving scenario and drivers' behaviour was quite similar during both tests (Kombi and Google Glass evaluation).

The average time for making the change is slightly lower using Kombi (0.2 seconds lower than using Google Glass). A graphical representation of the results is presented in Figure 6.7. Overall, a high resemblance between the elapsed times exists. In 5 cases the time for performing a gear change was smaller using a Google Glass unit and in 7 cases was higher with Google Glass. Results do not allow to assess the benefits or disadvantages of any display.

Table 6.3: Driving performance comparison for gear changing: Actual Gear (AG) to Recommended Gear (RG).

	KOMBI		Google Glass	
Participants	Avg. time for changing AG to RG	SD time for changing AG to RG	Avg. time for changing AG to RG	SD time for changing AG to RG
1	2,5	6,8	1,7	5,5
2	3,7	19,4	4,0	19,2
3	4,1	8,6	3,2	6,7
4	2,8	5,3	2,5	7,7
5	2,0	5,0	3,0	4,6
6	1,5	5,9	2,0	7,3
7	2,8	4,2	2,6	3,7
8	2,7	6,4	2,3	6,2
9	1,4	4,8	2,3	8,7
10	2,1	6,6	3,3	6,6
11	2,1	6,0	2,5	11,6
12	3,9	11,0	5,4	13,9
AVG.	2,7	7,5	2,9	8,5
STD. DEV	0,9	4,2	1,0	4,4

Figure 6.7: Time for changing AG to RG (HMD vs Kombi)



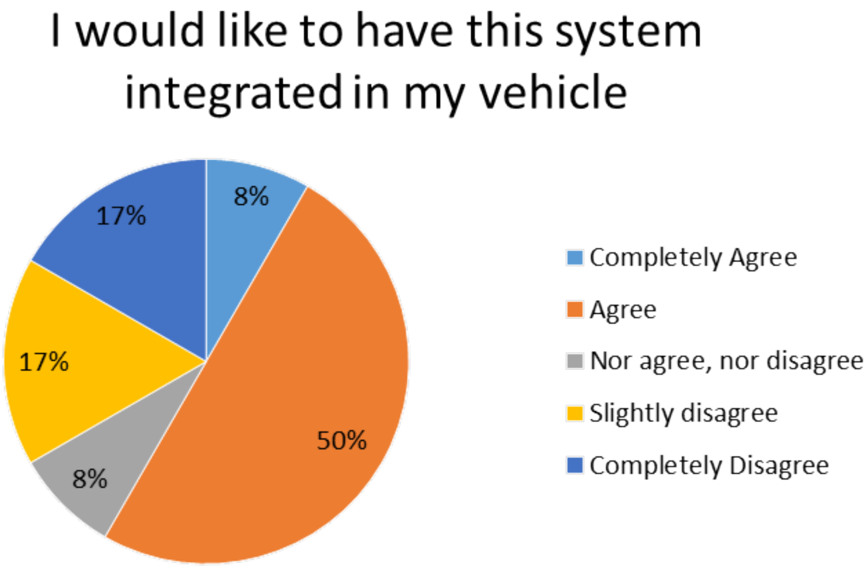
Participants' acceptance

Participants were asked for their opinion on the overall system considering all the designed applications; if they would use functionality proposed and if they would like to have this kind of system integrated in their vehicles. In relation to the first question; 58% of participants answered they would use the system frequently. Nevertheless they considered that the

Google Glass device is still immature and there are some pending issues to resolve as the battery life and the contrast of the display.

With respect to the integration of Google Glass in their vehicles (Figure 6.8), 58% of participants reported they would like to have the systems integrated in their vehicle. Nevertheless, they commented that the device may distract when used while driving depending on the type of functionality.

Figure 6.8: HMD Integration likeness



Participants were asked for their opinion in regards to implemented functionalities. They reported a positive feedback about the functionality *"Climate – Audio Source Setup"* (84% of participants exposed that would use the application often). Overall, this functionality was more accepted and enjoyed than *"Vehicle Status"*, which reported an appreciation of only 33%.

Participants were also asked if they would use the functionality *"Coaching"* and what is the utility level they see in this application. Overall, the application was considered very attractive and caused emotion on them; this emotion is understandable as it was their first experience using Google Glass. Nevertheless, they would not use this functionality in a daily basis, 75% of participants considers this functionality useful, but only a 42% of the sample would use it frequently.

In addition, some participants had never used the Kombi as an indicator to make gear changes; which means that the functionality itself is not used commonly (90 % of participants do not use the gear shift recommendation as a gear change indicator). Participants commented they would be interested in alerts and dynamic recommendations for notifying certain warnings as speed limit overpassed, entrance to highways, traffic, rain alerts.

Regarding participants' opinion in relation to the distraction level consequent for using a Google Glass device while driving, 75% of participants agreed that Google Glass causes distraction while driving. None relation between gender or appreciation by technology was detected in these results.

Due to the innovative nature of the HMD, the approval towards this type of interface may be conditioned; some participants showed certain predisposition before using the device for the first time. Therefore, some results may not reflect real objective appreciations. Also, the Google Glass unit was an immature technology and the display sometimes could not be seen properly, future improvements may enhance the presentation of information and more positive feedbacks may be given.

A Google Glass display for presenting information and interacting with applications while driving has not shown a negative impact on visual demand (glance pattern analysis). Nevertheless, it is clear that a improper use of the device would probably have great consequence on road safety, this effect can also occurs when using other interfaces available in the vehicle. Therefore, a good management of the information should be achieved when integrating a HMD in driving scenarios.

6.3 Summary

Since new gadgets and interfaces are being incorporated in people lives, vehicle manufacturers must evaluate the implications of using these technologies in a vehicle. In this regards, an extended set of principles regarding the integration of multiple devices in a driving scenario was proposed. The main goal is to improve the driving experience at the same time of increasing road safety. Proposed principles are an initial step to a complete framework that permits the categorization of interfaces in the vehicle.

As proof of concept, a HMD was integrated in a vehicle to validate its real benefits. Certain functionalities oriented to provide informative information or simple interaction were implemented in the device. Once developed, a usability and acceptance test was performed to evaluate the real benefits of this system. For this purpose, subjective perception and glance patterns were used as metrics. The section also presents some basic technical concept that can be used for integrating a Google Glass into a vehicle using IoT protocols as CoAP.

This study was very helpful for understanding the impact of using a Google Glass unit while driving and for getting user's opinions. The sample used for the test was small for finding patterns and correlating results depending on profile variables. Nevertheless, it was heterogeneous enough and useful for making a first analysis about the implementation of this type of wearables in a vehicle.

Google Glass, as an innovative product was considered attractive by some participants and rejected by others. During the test,certain predisposition towards to its implementation was observed; this inclination may have affected participants' objective opinion and altered their judgement when answering the questionnaire. For example, some participants considered the device very disturbing before using it for the first time and others that it was not intrusive at all.

The subjective analysis of driving workload did not offer conclusive results for determining the less demanding display (Glass or Kombi). Regarding objective results (glance pattern), participants had a lower glance duration when using the Google Glass in comparison to the Kombi. In relation to the acceptance, the overall first impression of Google Glass is positive, nevertheless people think that the product is still immature and its integration in driving scenarios must be well planned and supervised for avoiding their unwise use.

Chapter 7

Conclusions

Vehicle manufacturers are incorporating each time more services, functionalities and interfaces in vehicle infotainment systems. Its objective is to face the growing demand for new functions requested by users and to be at the forefront of offering inbound technologies. The integration of these new functionalities in vehicles is sometimes a straightforward process that brings advantages with respect to road safety and user experience; in other cases, the services are very complex and additional adjustments must be made to mitigate the effects on road safety. As a consequence of these adaptations, the purpose of a service is sometimes affected and functionalities offered are reduced.

When integrating new functionalities or interfaces into vehicles, much uncertainty exists about what it is the best strategy to do it. In most cases, goals achieved are the increment of road safety or improvement of user experience; but in others, applications are too complex and an not optimal design may bring with it some issues related to user experience.

This dissertation presents an approach for undertaking the integration of new interface technologies into vehicle infotainment systems, taking into account not only user experience but also, and more important, road safety. For this purpose, the project was addressed in two main parts; the first phase had as goal the creation of a reliable estimator of driving complexity for the current scenario, the second part was devoted to the compilation of existing design guidelines and recommendations into a new set of principles that could be used for creating adaptive interfaces in driving scenarios.

This final chapter presents not only an overall conclusion in regards of the two part of the project, but also exposes which were the main limitations and what future steps could be carried out. Overall, the project has shown positive results, and methodologies proposed appear to be adequate given the data collected. As a result, these results are expected to lead to future research in which the limitations presented here as well as a compilation of a greater amount of data can be used to improve the system.

In the first part of the project, a model capable of estimating driving complexity was built. For this purpose, several participants were recruited and asked them to drive an instrumented vehicle in real driving scenarios. While driving, participants were requested to respond to a PDT and indicate their perceived workload given as consequence of current driving complexity. Variables associated to the driving scenario and participants feedbacks were logged and analysed using data mining techniques.

As result, the more relevant variables useful for characterizing the driving complexity in terms of perceived workload and response time were identified. As previously exposed, the PDT has not shown to be an adequate mechanism for assessing driver workload given the test designed; response times had small variations for different scenarios and were correlated to some participants. Also, as the PDT event is randomly activated, some important driving conditions may have not been monitored. Future researches may evaluate users response time when the PDT event is activated in certain predefined situations for all of them in order to compare possible changes and variations.

On the contrary, the subjective workload perception of drivers seems to be a good indicator of driving complexity, many researches have validated the use of this metric as a reliable indicator of workload. This variable is easy to collect in tests and its value can expresses in a timely manner the perceived workload. Given the purpose followed; adapt user interfaces depending on driving complexity, four values of subjective perception were considered enough to be estimated, these were expressed as: “Very Low”, “Low”, “High” and “Very High”.

Several variables were selected to be used as inputs for making predictions, these variables were related to driving parameters, road type and environmental conditions. An important limitation of data collected was the lacking of profile information for creating the model, this variable was not incorporated into the final model given that the participants sample was not enough for making such categorization. Based on previous researches, it is expected that this variable has also an effect on driving complexity perception. Future researches are encouraged to incorporate parameters such as age, gender and physiological information and a larger sample of participants.

Once the most suitable variables for categorizing the driving complexity were identified, a model was constructed using machine learning methods. The use of machine-learning methods provide easy adaptability of the model when adding new inputs or fresh data. Furthermore, a some kind of predictive model is also required, since some scenarios may not been known in advance. Several methods were evaluated such as Multi-Layer Perceptron, Probabilistic Neural Networks, Decision Trees and Random Forest. From all these, the one that gave a satisfactory performance in terms of prediction accuracy and training time was Random Forest.

Once the model was built, it was incorporated into a vehicle in order to validate its performance in real time. After some tests, the system showed to be fast enough for making real time prediction and therefore is suitable for the purpose followed: adapt vehicle interfaces in driving scenarios according to the scenario complexity. As advantages, the proposed system makes use of variables that are already available and requires a low processing power which enables its integration into current vehicles without the need for high investment.

Given the predictor of driving complexity, infotainment systems can take several actions in regards of improving user experience and increasing road safety. Most of research undertaken for improving IVIS is oriented to define an optimal design that can be used in every scenario; nevertheless, driving is a very dynamic task and the workload required by the driver is constantly changing. In this regards, the design should also be dynamic and adapt the presentation and interaction with functionalities depending on the driving environment.

In this dissertation, a set of recommendations to be used for creating IVIS that can adapt their behaviour given a value of driving complexity are proposed. Taking as reference existing guidelines and recommendations with respect to HMI design, a new set of principles were compiled in 5 main concepts: Consistency, Modality, Adequate timing, Situation Awareness and Representation of information.

These set of principles were used for adapting an available HMI and testing its performance in real driving scenarios. Usability and acceptance test were executed recruiting participants and asking them to fulfil a set of questionnaires. Results shown very positive feedbacks in regards of incorporating this kind of systems into vehicles. Also, some benefits in terms of perceived safety are exposed. Further evaluations are proposed to be performed in order to certify implications trough empirical results.

In regards of easing the interaction with the infotainment system while driving, new interfaces such as HMD, HUD and smartwatch can also be added. A usability test was performed in order to evaluate the impact of using a HMD while driving for some applications. Results show that even when the device was considered dangerous to be used while driving, a reduction in the glance time compared to a commonly used interface (Kombi) was observed. As a conclusion, we expect that certain innovative technologies have problems being accepted by people as a consequence of some degree of predisposition. Also, the opposite may happen when certain technologies that are not suitable for use while driving shows great acceptance by people. The purpose of this section was to provide a first approach to the integration of portable devices such as wearables that are gaining each time more importance in people's lives.

Given the innovative approach of the research, several issues were found after the tests and many improvements can be achieved. This dissertation offers an initial step and further evaluations are expected to lead to additional conclusions. It is important to highlight that the methodology applied shown positive results and the additions of improvements proposed will surely increase the accuracy of the final model. A summary of what are considered the most important improvements is presented below.

As any mathematical analysis, a larger amount of data collected would probably contribute to improve the final model created. This new test can incorporate variables that were not measured due to cost and time limitations. Some examples are profile patterns, additional environmental variable, and previous knowledge of road topology. Also, interesting results could be obtained if a similar test is performed selecting road safety experts as participants for the tests. Results are expected to give conclusions about road safety expectations since many experts opinion would be compiled in one single model. This model can serve as basis for future road designs and traffic analysis.

In regards of the adaptive HMI, more tests can also be performed in order to certify results obtained. Also, the set of principles could be applied on another commercial HMI to validate results, and if possible find how these principles can be improved. A test oriented to evaluate the user appreciation after using the system for a long time can also offer interesting results to determine if users get used to the behaviour of the systems and larger benefits are achieved. Also, other interfaces can be tested as additional interfaces for showing information and interacting with functionalities.

Finally, one of the biggest advantages of this system is related to the simplicity in the integration in vehicles once the predictive model is built. The model make use of already available variables in vehicle and the accuracy obtained is adequate for the purpose followed, therefore the investment needed for its integration in production vehicles should be very low. Moreover, given the low computing power needed, current infotainment system should be capable of achieving real time predictions without changing the current hardware. With the integration of online features, the model can also be in constant improvement trough update releases.

As an answer to the hypothesis presented in the first section of the document, we can summarize the benefit of the research with the following sentence: *"Since we are able to monitor different parameter that allows computing an estimation of the* ***driving complexity****, and use this indicator for adapting what, when and how to show and interact with the* *information in multiple interfaces, we can improve* ***driving experience*** *and also probably increasing* ***road safety*** *if a right adaptive strategy is applied".*

Appendices

Appendix A

Road Crash Statistics

Table A.1: Vehicle accidents with victims in Spain (2014), Source: DGT (Dirección General de Tráfico (DGT) (2016))

TIPO DE ACCIDENTE	**VÍAS INTERURBANAS**		**VÍAS URBANAS**	
	Total	**Mortales**	**Total**	**Mortales**
Frontal	1356	13	2907	160
Fronto-lateral	13.659	42	17.837	148
Lateral	4.658	5	6.960	18
Por alcance	11.273	11	17.470	71
Múltiple o en caravana	2.496	5	4.901	36
Colisión contra obstáculo o elemento de la vía	1.662	6	2.256	14
Atropello a personas	10.952	142	11.724	241
Atropello a animales	45	0	469	1
Vuelco	1.411	5	2.375	14
Caída	2.606	7	3.415	25
Salida de la vía por la izquierda con colisión	411	14	2.424	91
Salida de la vía por la izquierda con despeñamiento	15	0	237	13
Salida de la vía por la izquierda con vuelco	60	2	1.356	52
Salida de la vía por la izquierda, otro tipo	175	4	1.167	29
Salida de la vía por la derecha con colisión	820	24	3.393	128
Salida de la vía por la derecha con despeñamiento	29	3	369	33
Salida de la vía por la derecha con vuelco	128	2	2.240	55
Salida de la vía por la derecha otro tipo	1.129	10	4.895	41
Otro tipo de accidente	3.538	50	5.175	159
T O T A L	**56.423**	**345**	**91.570**	**1.329**

Table A.2: Crash statistics in terms of light conditions in Spain (2014), Source: DGT (Dirección General de Tráfico (DGT) (2016))

Condición luz		VÍAS INTERURBANAS	VÍAS URBANAS	TOTAL	TOTAL (%)
Pleno día	Accidentes con víctimas	24826	41046	65872	70,91
	Accidentes mortales	607	203	810	0,87
Crepúsculo	Accidentes con víctimas	1864	3147	5011	5,39
	Accidentes mortales	53	20	73	0,08
Noche, via suficientemente iluminada	Accidentes con víctimas	2433	11547	13980	15,05
	Accidentes mortales	53	103	156	0,17
Noche, vía insuficientemente iluminada	Accidentes con víctimas	411	317	728	0,78
	Accidentes mortales	13	1	14	0,02
Noche, vía no iluminada	Accidentes con víctimas	5613	366	5979	6,44
	Accidentes mortales	258	18	276	0,30
Total		36131	56768	92899	100

Appendix B

Machine Learning Methods

Figure B.1: Machine Learning Map.

Source: https://jixta.wordpress.com/2015/07/17/machine-learning-algorithms-mindmap/

Table B.1: Accuracy and training time for PNN depending on theta values. (Knime)

Dataset	ThetaPlus	Thetaminus	Avg. Accuracy [%]	Avg. Training Time [ms]
A	0,1	0,3	89 ± 0	20 ± 9
A	0,1	0,5	56 ± 0	21 ± 10
A	0,1	0,7	41 ± 0	15 ± 1
A	0,1	0,9	30 ± 0	16 ± 1
A	0,2	0,1	100 ± 0	21 ± 9
A	0,2	0,3	93 ± 0	36 ± 9
A	0,2	0,5	67 ± 0	26 ± 9
A	0,2	0,7	33 ± 0	21 ± 9
A	0,2	0,9	30 ± 0	16 ± 1
A	0,3	0,1	100 ± 0	21 ± 9
A	0,3	0,3	93 ± 0	26 ± 18
A	0,3	0,5	67 ± 0	16 ± 1
A	0,3	0,7	41 ± 0	26 ± 9
A	0,3	0,9	30 ± 0	239 ± 374
A	0,4	0,1	100 ± 0	26 ± 9
A	0,4	0,3	89 ± 0	31 ± 1
A	0,4	0,5	70 ± 0	26 ± 9
A	0,4	0,7	37 ± 0	26 ± 9
A	0,4	0,9	30 ± 0	16 ± 1
A	0,5	0,1	100 ± 0	21 ± 9
A	0,5	0,3	89 ± 0	21 ± 9
A	0,5	0,5	70 ± 0	15 ± 1
A	0,5	0,7	52 ± 0	21 ± 9
A	0,5	0,9	30 ± 0	21 ± 9
A	0,6	0,1	100 ± 0	26 ± 9
A	0,6	0,3	96 ± 0	47 ± 41
A	0,6	0,5	74 ± 0	26 ± 9
A	0,6	0,7	48 ± 0	21 ± 9
A	0,6	0,9	33 ± 0	26 ± 10
A	0,7	0,1	100 ± 0	26 ± 9
A	0,7	0,3	96 ± 0	31 ± 1
A	0,7	0,5	74 ± 0	31 ± 1
A	0,7	0,7	48 ± 0	26 ± 9
A	0,7	0,9	33 ± 0	20 ± 9
A	0,8	0,1	100 ± 0	26 ± 9
A	0,8	0,3	96 ± 0	26 ± 9
A	0,8	0,5	74 ± 0	20 ± 9
A	0,8	0,7	48 ± 0	21 ± 9
A	0,8	0,9	37 ± 0	26 ± 9
A	0,9	0,1	100 ± 0	21 ± 9
A	0,9	0,3	96 ± 0	21 ± 10
A	0,9	0,5	74 ± 0	26 ± 10
A	0,9	0,7	56 ± 0	21 ± 9

A	0,9	0,9	30 ± 0	20 ± 9
A	1	0,1	100 ± 0	26 ± 9
A	1	0,3	96 ± 0	21 ± 9
A	1	0,5	74 ± 0	21 ± 9
A	1	0,7	56 ± 0	26 ± 9
A	1	0,9	37 ± 0	21 ± 9
B	0,1	0,3	70 ± 0	31 ± 1
B	0,1	0,5	70 ± 0	26 ± 10
B	0,1	0,7	70 ± 0	31 ± 16
B	0,1	0,9	70 ± 0	31 ± 1
B	0,2	0,1	70 ± 0	36 ± 9
B	0,2	0,3	70 ± 0	31 ± 0
B	0,2	0,5	70 ± 0	26 ± 9
B	0,2	0,7	70 ± 0	26 ± 9
B	0,2	0,9	70 ± 0	31 ± 1
B	0,3	0,1	70 ± 0	26 ± 9
B	0,3	0,3	70 ± 0	26 ± 9
B	0,3	0,5	70 ± 0	26 ± 9
B	0,3	0,7	70 ± 0	26 ± 9
B	0,3	0,9	70 ± 0	26 ± 9
B	0,4	0,1	70 ± 0	21 ± 9
B	0,4	0,3	70 ± 0	26 ± 9
B	0,4	0,5	70 ± 0	31 ± 16
B	0,4	0,7	70 ± 0	26 ± 9
B	0,4	0,9	70 ± 0	31 ± 1
B	0,5	0,1	70 ± 0	26 ± 9
B	0,5	0,3	70 ± 0	26 ± 9
B	0,5	0,5	70 ± 0	31 ± 0
B	0,5	0,7	70 ± 0	37 ± 18
B	0,5	0,9	70 ± 0	26 ± 9
B	0,6	0,1	70 ± 0	26 ± 9
B	0,6	0,3	70 ± 0	26 ± 10
B	0,6	0,5	70 ± 0	36 ± 9
B	0,6	0,7	70 ± 0	31 ± 0
B	0,6	0,9	70 ± 0	36 ± 9
B	0,7	0,1	70 ± 0	31 ± 1
B	0,7	0,3	70 ± 0	21 ± 9
B	0,7	0,5	70 ± 0	21 ± 9
B	0,7	0,7	70 ± 0	26 ± 9
B	0,7	0,9	70 ± 0	36 ± 24
B	0,8	0,1	70 ± 0	26 ± 10
B	0,8	0,3	70 ± 0	26 ± 9
B	0,8	0,5	70 ± 0	31 ± 1
B	0,8	0,7	70 ± 0	20 ± 9
B	0,8	0,9	70 ± 0	31 ± 1
B	0,9	0,1	70 ± 0	31 ± 1
B	0,9	0,3	70 ± 0	16 ± 1
B	0,9	0,5	70 ± 0	31 ± 0

B	0,9	0,7	70 ± 0	21 ± 9
B	0,9	0,9	70 ± 0	27 ± 9
B	1	0,1	70 ± 0	21 ± 9
B	1	0,3	70 ± 0	31 ± 0
B	1	0,5	70 ± 0	26 ± 10
B	1	0,7	70 ± 0	31 ± 16
B	1	0,9	70 ± 0	26 ± 9

Appendix C

Head Mounted Display Tables

Table C.1: Questionnaire: Technology Appreciation.

Las siguientes preguntas están relacionadas con la tecnología.					
Por favor, lea las siguientes afirmaciones y conteste en qué grado está de acuerdo.					
Considero que. . .	Muy en desacuerdo	Ligeramente en desacuerdo	Ni de acuerdo, ni desacuerdo	Ligeramante de acuerdo	Muy de acuerdo
1. la tecnología hace que todo funcione mejor					
2. el mundo sería un lugar mejor sin la tecnología					
3. debería haber más educación relacionada con la tecnología					
4. la tecnología es aburrida					
5. la tecnología es la base del futuro					
6. el uso de la tecnología hace que los países sean menos prósperos					
7. la tecnología es buena para el futuro de un país					
8. la tecnología aporta más cosas malas que buenas					
9. la tecnología es importante en la vida diaria					
10. la tecnología provoca desempleo					
Compruebe una vez más que ha contestado a todas las preguntas del cuestionario. ¡Muchas gracias!					

Table C.2: Workload Questionnaire (Subjective Perception)

Cuestionario carga mental	
Durante el recorrido que acabas de realizar, es posible que hayas sentido algunas limitaciones y dificultades respecto al modo en que sueles conducir. Por esta razón, nos proponemos evaluar estas posibles modificaciones a través de 5 factores. Estos factores son descritos en la siguiente tabla. No dudes en consultar cualquier cuestión si lo necesitas.	

Factor	**Descripción**
Demanda de atención global	Demanda mental (pensar a cerca de algo, decidir. . .), requerida durante el test para realizar la actividad entera.
Demanda visual	Demanda visual requerida durante el test para cambiar el enfoque del display a la vía.
Estrés	Nivel de estrés durante toda la actividad tal como fatiga, sentimiento de inseguridad, irritación, desánimo/ desaliento.
Demanda temporal	Presión y sentimiento específico de restricción relacionado con la demanda de tiempo durante el transcurso de toda la actividad.
Interferencia	Interrupción del estado del conductor y sus consecuencias en la conducción. Como siente que ha afectado su conducción.

Para cada factor, vas a valorar el nivel de esfuerzo que has sentido durante la sesión en una escala que va de 0 (bajo) a 5 (alto) respecto al modo en que sueles conducir.

		0	1	2	3	4	5	
1. Demanda de atención global	baja							alta
2. Demanda visual	baja							alta
5. Estrés	baja							alta
6. Demanda temporal	baja							alta
7. Interferencia	baja							alta

Evalúe la conducción que acaba de hacer en comparación a la conducción habitual.

Peor				Igual					Mejor
1	2	3	4	5	6	7	8	9	10

Appendix D

OSM Usage

The road type variable is extracted from a public database called Open Street Maps (OSM), here the information is stored in a "*.OSM" format, or in its compressed version "*.OSM.BPF". More information regarding OSM is availables in a Wiki (OpenStreetMap Wiki (2014)). In OSM the core element of the database are nodes, these are classified in:

- Relation: Consists of an ordered list of one or more ways and/or relations which is used to define logical or geographic relationships between other elements.
- Way: Ordered list of nodes which normally also has at least one tag or is included within a relation. A way can have between 2 and 2,000 nodes. Ways can be open or closed (the last node on the way is also the first on that way).

For easing the access from external applications the OSM format was translated to a sql database. In the final database, two tables contained all the information, one allowed to extract the node id given the GPS coordinates, and the other table use this node id to get information about node type, maximum speed allowed and road type. The procedure followed consisted on:

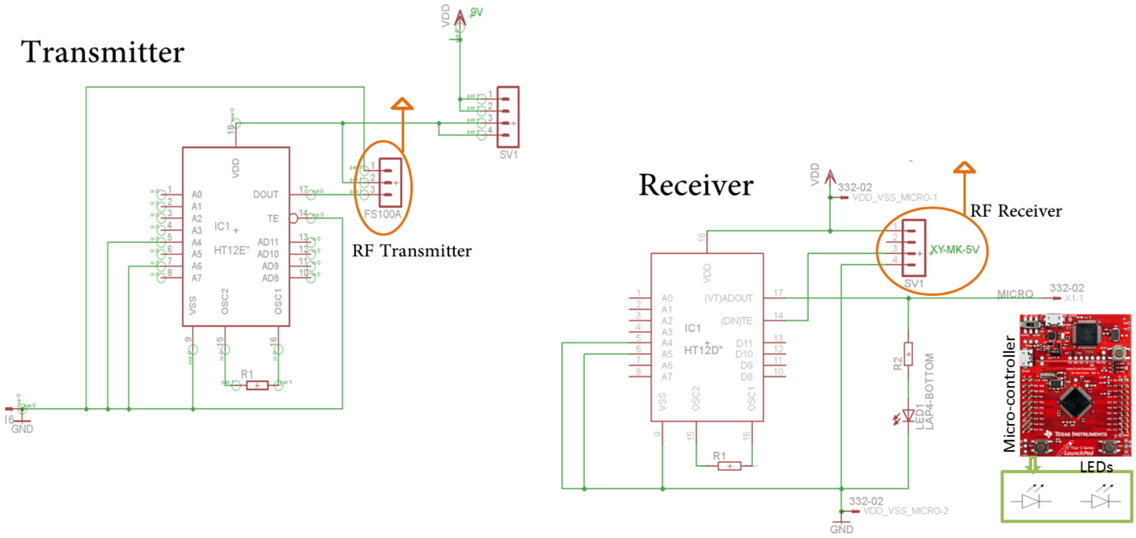
1. Download *.osm.pbf file.
2. Convert *.osm to *.db3 (sqlite using the application OsmPbf2sqlite).Some information about the database:
   (a) Node has the references of nodes to Lat,Long
   (b) KeyNode have the relation between Integers and meaning (Name, WayType, date, source,etc), this is linked to an INT.
   (c) KeyValueWay has the information of the way once known the way_id.
   (d) idKey(From KeyNode Table) idVal (From ValueNode Table) are the associations.
3. Filter database removing unnecessary roads or areas outside test boundary.
4. Created tables in order to simplify data structure and allow make faster queries responses.

Appendix E

PDT Micro

The Peripheral Detection Task (PDT) installed in the vehicle was composed by a micro-controller, two LEDs, a Radio Frequency (RF) transmitter, a RF receptor and two pushable buttons. Figure E.1 shows a diagram of the implemented system which is composed by two main boxes, a Transmitter box located in the steering wheel and a Receiver box connected to the Linux machine. The Transmitter consists of a circuit, a 9V battery and pair of pushable buttons that once pressed, they power up the electronic components and initiate the transmission of a RF signal. For this purpose, it is used a encoder (HT12-E) and a RF transmitter (FS100A) that works at 433MHz.

Figure E.1: PDT Diagram.



The micro-controller is the main core of the system and have the following tasks: (1) turn on two LEDs periodically in a random time, (2) detect any incoming signal received trough the RF receptor (XY-MK-5V) and (3) transmit trough USB/UART the response time to a Linux machine. The receiver block is powered directly from a the USB connection with the Linux machine.

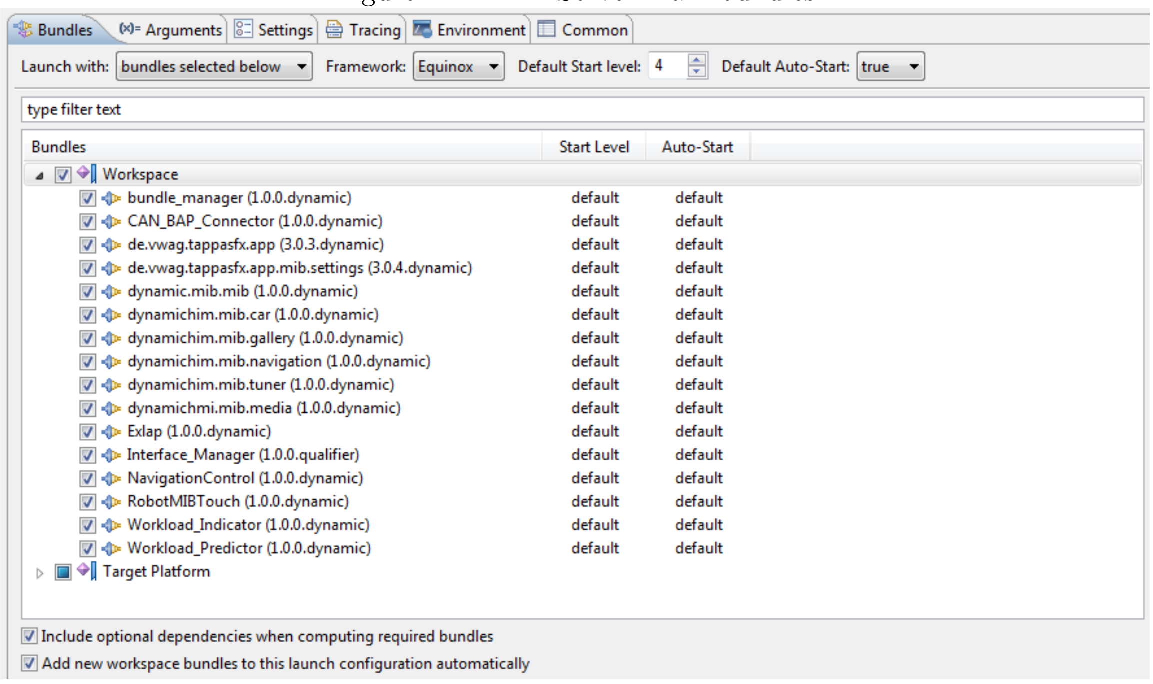
Appendix F

HMI Server implementation

The HMI Server was developed using Java as programming language and Eclipse as Integrated Development Environment. The framework used for the managing of services was OSGI Equinox and the one used for creating the user interface of the project JavaFX.

The main structure of the project consists in a set of bundles that are initialized in a default priority according to its dependencies. The list of main bundles is presented in the Figure F.1. Bundles starting with the prefix *dynamichmi* or *de.vwag* are associated to user interfaces. Bundles *CAN_BAP_Connector*, *Exlap* and *RobotMIBTouch* are used for implementing communication interfaces with the vehicle. The adaptive nature of the system is implemented by the bundles *bundle_manager*, *Interface_Manager*, *Workload_Indicator* and *Workload_Predictor*. The *NavigationControl* bundle implements the navigation logic for calculating routes and informing about maneouvers.

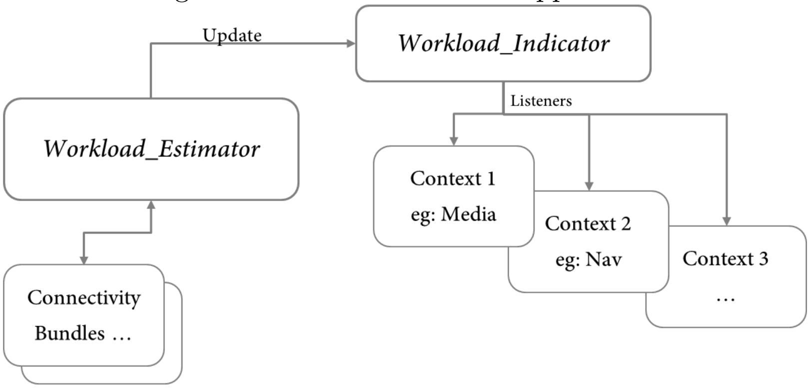
Figure F.1: HMI Server main bundles



An structure of the application is presented in Figure F.2. Basically, the *Workload_Estimator* makes periodic computations about the driving complexity and report the values to the

Workload_Indicator. User interfaces context are subscribed to this bundle and any change is immediately received by the context in order to apply proper changes. Connectivity Bundles are related to vehicle communication and profile information.

Figure F.2: Structure of the Application



The main class of the project in charge of starting the User Interface application and initiating the communication with the rest of bundles is the Main.java class located in the *InterfaceManager* bundle. The class extends the *Application* interface in a JavaFX environment in order to start a user application. An explanation of the class implementation is presented in the code below where the comments expose the meaning of the actions undertaken.

```
package main;

import java.io.File;
import java.io.IOException;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import javafx.application.Application;
import javafx.application.Platform;
import javafx.event.EventHandler;
import javafx.fxml.FXMLLoader;
import javafx.scene.Scene;
import javafx.scene.image.Image;
import javafx.scene.layout.AnchorPane;
import javafx.stage.Stage;
import javafx.stage.StageStyle;
import javafx.stage.WindowEvent;
import car_communication.CarConnection;
import car_communication.Get_Driving_scenario;
import car_communication.Get_Driving_scenario2;
import Logger.Global_Logger;
import Logger.Profile_Logger;
import Logger.Profile_Logger_Normalized;
import screen.VariableTest_Controller;
import screen.VariableViewerController;
import screen.VariableViewerControllerSimp;
import screen.loading_Controller;
import tcp_comm.Server;
import uart.UARTPort;
import uart.randomPDT;
```

```
import workload.Workload_manager;
import workload.Workload_managerPMML;
import database.DatabaseManager;

/**
 * Main Thread that initiates the GUI Application and the rest of threads dedicated to
     communication and workload computation.
 * */
public class Main extends Application implements Runnable{

  //Variable used to indicate the state of the program.
  public volatile static boolean running = true;
  private static boolean testing = true;

  //MAIN FOR TESTING ONLY
  public static void main(String[] args) throws Exception {
    launch(); //It is allowed just one Application for VM, In case of requiring new
        Windows, add another Stage.
  }

  //In case of being initialized as Runnable
  @Override
  public void run() {
    launch();
    System.out.println("Started Main Thread");
  }

  @Override
  public void start(Stage primaryStage) {

    //Indicates that the System is loading.
    Variables.isloading = true;

    //Gets information about the Operative System.
    Variables.OS = System.getProperty("os.name");
    System.out.println("Starting Program!! Running on OS: "+Variables.OS);

    //Method that Create and Register Variables.
    Variables.CreateVariables();

    //Open or create profile database and table if it does not exist.
    Thread loadDB = new Thread(){public void run(){
      DatabaseManager db_manager = new DatabaseManager();
      db_manager.createDatabase();
      db_manager.createtable_profiles();
    }};loadDB.start();

    //Database loading complete.
    //loadingcontroller.close();

    //Show Profile Dialog (Create a New User or Load an existing one).
    if(testing){
      NewWindow("Profile Selection", "/screen/Profile.fxml",true,false,true);
    }else{
      NewWindow("Profile Selection", "/screen/Profile_simplified.fxml",true,false,true);
    }

    // In devMode, creates a monitoring window of variables.
    if(testing){
```

```
    //Monitoring Window.
    FXMLLoader Monitoring = NewWindow("Monitoring",
        "/screen/VariablesViewer.fxml",false,true,true);
    VariableViewerController Monitoringcontroller = Monitoring.getController();
    Monitoringcontroller.setMainApp(this);
  }else{
    //Monitoring Window (Simplified).
    FXMLLoader MonitoringSimplified = NewWindow("Monitoring",
        "/screen/VariablesViewerSimplified.fxml",false,true,true);
    VariableViewerControllerSimp MonitoringcontrollerSimplified =
        MonitoringSimplified.getController();
    MonitoringcontrollerSimplified.setMainApp(this); //Pass the Class as parameter for
        interacting from the controller.
  }

  //Thread initialization for communicating with the Micro-Controller (UART)
  Thread UART_Comm = new UARTPort();
  UART_Comm.start();

  //Timer Task initialization for PDT Random sending (start(min,max)) min and max values
      (seconds) for random times in PDT led.
  randomPDT PDT = new randomPDT();
  PDT.start(45,120);

  //Thread initialization for Reading Vehicle Data (Kayak library used).
  CarConnection CarComm = new CarConnection();
  CarComm.start();

  //Computes driving scenario and updates related variables.
  Get_Driving_scenario ds = new Get_Driving_scenario();
  ds.start();

  //Mobile Subjective Connection to Mobile (TCP Comm.)
  Server MobileConn = new Server();
  MobileConn.start();

  //Create LOG Folder
  File dir = new File("LOG");
  dir.mkdir();

  //Thread initialization used for Logging variable data during road.
  Variables.StarttimeLog = new
      SimpleDateFormat(Integer.toString(Variables.SubjectID)+"__yyyyMMdd_HHmm").
          format(Calendar.getInstance().getTime());
  Global_Logger Profile_Global_Logger = new
      Global_Logger(System.getProperty("user.dir")+"/LOG/"+Variables.StarttimeLog);
  Profile_Global_Logger.start();

  //Thread initialization for estimating the Workload. PMML Model.
  Workload_managerPMML wm = new Workload_managerPMML();
  wm.start();

  //Indicates that the System has already loaded and started every Thread.
  Variables.isloading = false;

  //Start Counting running time.
  Variables.runningTime = System.currentTimeMillis();
}
```

```
//Function than may be called by others to present the final Questionnaire and later
    close the Application.
public void close() {

  //Indicate other Threads that the Application should be closed.
  running = false;

  //Show window presenting a Final Questionnaire.
  NewWindow("Questionnaire", "/screen/Questionnaire.fxml",true,true,true);

  //Suspend Application.
  System.exit(0);
}

//Function than may be called by others to close the Application without presenting the
    Final Questionnaire.
public void abort() {

  //Indicate other Threads that the Application should be closed.
  running = false;

  //Suspend Application.
  System.exit(0);
}


/**
 * Initializes Windows.
 * @param title Title of the stage.
 * @param Resource Resource location (fxml file)
 * @param ShowAndWait
 * @param overrided_close Indicates if it is possible to close the window from the
     button X.
 * @param decoration Indicates if it shows decoration (maximize, minimize and close
     buttons)
 * @return FXMLLoader
 */
public FXMLLoader NewWindow(String title, String Resource, boolean ShowAndWait, boolean
    overrided_close, boolean decoration) {

  FXMLLoader loader = new FXMLLoader();
  try {
    // Load root layout from fxml file.
    loader.setLocation(Main.class.getResource(Resource));
    AnchorPane rootLayout = (AnchorPane) loader.load();
    FXMLLoader.load(getClass().getResource(Resource));

    // Show the scene containing the root layout.
    Stage stage = new Stage();
    stage.getIcons().add(new Image("shell_icon.png"));
    Scene scene = new Scene(rootLayout);
    stage.setOnCloseRequest(new EventHandler<WindowEvent>() {
      @Override
      public void handle(WindowEvent event) {
        Platform.exit();
        abort();
      }
    });
```

```
      if(overrided_close){
        stage.setOnCloseRequest(new EventHandler<WindowEvent>() {
          @Override
          public void handle(WindowEvent event) {
            event.consume();
          }
        });
      }

      if(!decoration){
        stage.initStyle(StageStyle.UNDECORATED);
      }
      stage.setScene(scene);
      stage.setResizable(true);
      if(ShowAndWait){
        stage.showAndWait();
      }
      else{
        stage.show();
      }
    } catch (IOException e) {
      e.printStackTrace();
    }
    return loader;
  }

}
```

The Bundle responsible of managing changes to user interface context is the *Workload_Indicator*. Any Context can access this bundle in order to get information about the current workload, while in the same manner the *Workload_Estimator* bundle can update the current complexity directly there. An implementation of the main class of this bundle is presented below.

```
/**
 *
 * Basic Class to provide a global access to Workload Value from different bundles.
 * The access to the workload indicator can be done using JavafxBeansProperties.
 *
 * */
public class Workload {

  static boolean DefaulInitDynStatus = true;

  static volatile int Workload;

  private static IntegerProperty workloadProperty = new SimpleIntegerProperty();

  private static BooleanProperty ActiveDynHMIPropertyProperty = new
      SimpleBooleanProperty(DefaulInitDynStatus);

  // Get current driving complexity
  public static int Get(){
    return Workload;
  }
```

```
  // Set/Update driving complexity
  public static void Set(int workload){
    this.Workload = workload;
    workloadProperty.set(workload);
  }

  // Return Workload Property (JavaFX method)
  public static IntegerProperty GetWorkloadProperty(){
    return workloadProperty;
  }

  // Return Adaptive status (Boolean -> On/Off) (JavaFX method)
  public static BooleanProperty GetDynamicSystemStatus(){
    return ActiveDynHMIPropertyProperty;
  }

  // Set Adaptive status (Boolean -> On/Off) (JavaFX method)
  public static void SetDynamicSystemStatus(boolean activation_status){
    ActiveDynHMIPropertyProperty.set(activation_status);
  }

}
```

Bibliography

***Cited web pages were available and accessible on date 20/10/2017.**

Al-hegami, A. S. (2004). Subjective Measures and their Role in Data Mining Process. *In Proceedings of the 6th International Conference on Cognitive Sytems.*

Allport, D. A., Antonis, B., and Reynolds, P. (1972). On the division of attention: A disproof of the single channel hypothesis. *Quarterly Journal of Experimental Psychology*, 24(2):225–235.

Amditis, A., Andreone, L., Polychronopoulos, A., and Engström, J. (2005). Design and development of an adaptive integrated driver-vehicle interface: overview of the AIDE project. *IFAC Proceedings Volumes*, 38(1):103–108.

Amditis, A., Pagle, K., Joshi, S., and Bekiaris, E. (2010). Driver-Vehicle-Environment monitoring for on-board driver support systems: lessons learned from design and implementation. *Applied ergonomics*, 41(2):225–35.

Andrey, J., Mills, B., and Vandermolen, J. (2001). Weather information and road safety. *Institute for Catastrophic Loss Reduction, Toronto, Ontario, Canada*, (15).

Apple (2016). CarPlay.

Assistant, D. (2014). Mathematical Study OF Adaptive Genetic Algorithm (AGA) with Mutation and Crossover probabilities. 3(V):765–768.

Baber, C. (1994). Generic intelligent driver support: a comprehensive report of {GIDS}. *Applied Ergonomics*, 25(3):195–196.

Banfield, R. E., Hall, L. O., and Bowyer, K. W. (2007). A comparison of Decision Tree Ensemble Creation techniques. 29(1):173–180.

Beatty, J. (1982). Beatty Pupil Review PsychBull 1982.pdf. *Psychological bulletin*, 91(2):276–292.

Beirness, D. J. (1993). Do we really drive as we live? The role of personality factors in road crashes. *Alcohol, Drugs & Driving.*

Berthold, M. R., Cebron, N., Dill, F., Gabriel, T. R., Kötter, T., Meinl, T., Ohl, P., Sieb, C., Thiel, K., and Wiswedel, B. (2007). {KNIME}: The {K}onstanz {I}nformation {M}iner. In *Studies in Classification, Data Analysis, and Knowledge Organization (GfKL 2007).* Springer.

Berthold, M. R. and Diamond, J. (1998). Constructive training of probabilistic neural networks. *Neurocomputing*, 19(1-3):167–183.

Borges, L. C., Marques, V. M., and Bernardino, J. (2013). Comparison of data mining techniques and tools for data classification. *Proceedings of the International C* Conference on Computer Science and Software Engineering - C3S2E '13*, (August 2016):113.

Boys, R. (2004). CAN : Controller Area Network Introduction and Primer. 8:1–8.

Breiman, L. (1996). Bagging Predictors. *Machine Learning*, 24(421):123–140.

Breiman, L. (2001a). Random Forests. *Machine learning*, 45.1:5–32.

Breiman, L. (2001b). Statistical Modeling: The Two Cultures. *Statistical Science*, 16(3):199–215.

Broadbent, D. E. (1952). Listening to One of Two Synchronous Messages. *Journal of Experimental Psychology*, 44(1):51.

Brookhuis, K. a. and De Waard, D. (2010). Monitoring drivers' mental workload in driving simulators using physiological measures. *Accident Analysis and Prevention*, 42:898–903.

Brown, I. D. and Groeger, J. A. (1988). Risk perception and decision taking during the transition between novice and experienced driver status. *Ergonomics*, 31(4):585–597.

Brown, T., Marshall, D., Moeckli, J., and Smyser, T. (2007). A Final Report of SAfety Vehicle(s) using adaptive Interface Technology (SAVE-IT) Program: Task 14a Evaluation. Technical Report November.

Burns, P. C., Knabe, E., and Tevell, M. (2000). Driver Behavioral Adaptation to Collision Warning and Avoidance Information. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 44(20):3–315.

Car Connectivity Consortium (2016). MirrorLink.

Chai, T. and Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)? -Arguments against avoiding RMSE in the literature. *Geoscientific Model Development*, 7(3):1247–1250.

Christensen, L. (2013). Post-harmonised European National Travel Surveys.

Co-operation, O. F. O. R. E. (2007). Policy Brief. *Technology*, (February).

Commission of the European Communities (2008). ESoP - European Statement of Principles on human-machine interface. Technical report, Commission of the European Communities.

Copsey, S., Christie, N., and Drupsteen, L. (2011). *A review of accidents and injuries to road transport drivers. European Agency for Safety and Health at Work (EU-OSHA).*

Cortes, C. and Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, 20(3):273–297.

Crammer, K. and Singer, Y. (2001). On The Algorithmic Implementation of Multiclass Kernel-based Vector Machines. *Journal of Machine Learning Research (JMLR)*, 2:265–292.

Daiss-fechner, N. and Boat, J. (2005). Sight or Sound; A Study of Reaction Times.

De Waard, D. (1996). *The Measurement of Drivers ' Mental Workload*, volume 39. Groningen University, Traffic Research Center.

De Waard, D., Kruizinga, A., and Brookhuis, K. A. (2008). The consequences of an increase in heavy goods vehicles for passenger car drivers' mental workload and behaviour: A simulator study. *Accident Analysis & Prevention*, 40(2):818–828.

Dea, C., Heckler, M., Grunwald, G., Pereda, J., and Phillips, S. (2014). *JavaFX 8: Introduction by Example*. Apress, Berkely, CA, USA, 2nd edition.

Deery, H. A. (2013). Reprint of "Hazard and Risk Perception among Young Novice Drivers". *Journal of Safety Research*, 30(4):225–236.

Demšar, J., Curk, T., Erjavec, A., Gorup, Č., Hočevar, T., Milutinovič, M., Možina, M., Polajnar, M., Toplak, M., Starič, A., Štajdohar, M., Umek, L., Žagar, L., Žbontar, J., Žitnik, M., and Zupan, B. (2013). Orange: Data Mining Toolbox in Python. *Journal of Machine Learning Research*, 14:2349–2353.

Dewar, R. E. and Olson, P. L. (2002). Human Factors in Traffic Safety.

Dietterich, T. G. (2000). An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees. *Machine Learning*, 40:139–157.

Dirección General de Tráfico (DGT) (2016). Estadisticas España 2014 Trafico.

Dodd, D. K. and Mills, L. (1985). FADIS: A measure of the fear of accidental death and injury. *The Psychological Record*, 35(2):269.

Egeth, H. E. and Yantis, S. (1997). Visual attention: control, representation, and time course. *Annual review of psychology*, 48:269–297.

Elander, J., West, R., and French, D. (1993). Behavioral correlates of individual differences in road-traffic crash risk: an examination method and findings. *Psychological bulletin*, 113(2):279–294.

Engström, J., Monk, C. a., Hanowski, R. J., Horrey, W. J., Lee, J. D., McGehee, D. V., Regan, M. a., Stevens, a., Traube, E., Tuukkanen, M., Victor, T., and Yang, D. (2013). A Conceptual Framework and Taxonomy for Understanding and Categorizing Driver Inattention.

European Commission (2015a). Annual Accident Report 2015. Technical report, European Commission.

European Commission (2015b). Transport in Figures'–Statistical pocketbook 2015. Technical report, European Commission, Luxembourg.

European Commission, D.-G. T. and Energy (2009). Speeding. *European Commission, Directorate-General Transport and Energy.*

European Commission Project: DRIVE II (1993). Application of a Real-time Intelligent Aid for Driving and Navigation Enhancement (ARIADNE). Project V2004/14. Technical report, European Commission., Belgium.

European Transport Accident Statistics Working Party (2003). *Transport Safety Performance.*

Freund, Y. and Schapire, R. R. E. (1996). Experiments with a New Boosting Algorithm. *International Conference on Machine Learning*, pages 148–156.

Garay-Vega, L., Pradhan, A. K., Weinberg, G., Schmidt-Nielsen, B., Harsham, B., Shen, Y., Divekar, G., Romoser, M., Knodler, M., and Fisher, D. L. (2010). Evaluation of different speech and touch interfaces to in-vehicle music retrieval systems. *Accident Analysis & Prevention*, 42(3):913–920.

Gartner, N. H., Messer, C. J., and Rathi, A. K. (1997). *Monograph on traffic flow theory.* Federal Highway Administration.

Gitlin, J. M. (2014). The past, present, and future of in-car infotainment.

Google (2016). Android Auto.

Grane, C. and Bengtsson, P. (2013). Driving performance during visual and haptic menu selection with in-vehicle rotary device. *Transportation Research Part F: Traffic Psychology and Behaviour*, 18:123–135.

Green, S. (2000). Sleepy, groggy, angry: Drinking isn't the only road to impairment. *Traffic Safety (Chicago)*, (1).

Gruhn, P. (2011). Human machine interface (HMI) design: the good, the bad, and the ugly (and what makes them so). In *66th Annual Instrumentation Symposium for the Process Industries*, pages 27–29.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The WEKA Data Mining Software: An Update. *SIGKDD Explor. Newsl.*, 11(1):10–18.

Harms, L. and Patten, C. (2003). Peripheral detection as a measure of driver distraction. A study of memory-based versus system-based navigation in a built-up area. *Transportation Research Part F: Traffic Psychology and Behaviour*, 6(1):23–36.

Harper, R. P. and Cooper, G. E. (1986). Handling qualities and pilot evaluation. *Journal of Guidance, Control, and Dynamics*, 9(5):515–529.

Hart, S. G. and Staveland, L. E. (1988). Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In Hancock, P. A. and Meshkati, N., editors, *Human Mental Workload*, volume 52 of *Advances in Psychology*, pages 139–183. North-Holland.

Heaton, J. (2015). Encog: Library of Interchangeable Machine Learning Models for Java and C#. *Journal of Machine Learning Research*, 16:1243–1247.

Highway, N. and Safety, T. (2008). National Motor Vehicle Crash Causation Survey Report to Congress. (July).

Ho, C. and Spence, C. (2012). *The multisensory driver: Implications for ergonomic car interface design*. Ashgate Publishing, Ltd.

Hoedemacker, M., Ridder, S., and Janssen, W. (2002). Revier of European Human Factors Research on Adaptive Interface Technologies for Automobiles. Technical report, TNO Human Factors.

Hofmann, M. and Klinkenberg, R. (2013). *RapidMiner: Data Mining Use Cases and Business Analytics Applications*. Chapman & Hall/CRC.

Hoogendoorn, R. and Van Arem, B. (2013). Driver workload classification through neural network modeling using physiological indicators. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, (Itsc):2268–2273.

Hs, D. O. T. (2013). Research Note 2012 Motor Vehicle Crashes : Overview. (November).

Igel, C. and Hüsken, M. (2000). Improving the Rprop learning algorithm. *Proceedings of the Second International Symposium on Neural Computation*, pages 115–121.

Igel, C. and Hüsken, M. (2003). Empirical evaluation of the improved Rprop learning algorithms. *Neurocomputing*, 50:105–123.

Institute for Road Safety Research (2012). SWOV Fact sheet (The influence of weather on road safety). Technical Report February, Leidschendam.

Jahn, G., Oehme, A., Krems, J. F., and Gelau, C. (2005). Peripheral detection as a workload measure in driving: Effects of traffic complexity and route guidance system use in a driving study. *Transportation Research Part F: Traffic Psychology and Behaviour*, 8:255–275.

Jakus, G., Dicke, C., and Sodnik, J. (2015). A user study of auditory, head-up and multi-modal displays in vehicles. *Applied Ergonomics*, 46, Part A:184–192.

Jović, a., Brkić, K., and Bogunović, N. (2014). An overview of free software tools for general data mining. *37th International Convention MIPRO ...*, (May):26–30.

Jr, R. P. H. and Cooper, G. E. (1984). Handling Qualities and Pilot Evaluation 1984. pages 1–39.

Kong, J., Zhang, W. Y., Yu, N., and Xia, X. J. (2011). Design of human-centric adaptive multimodal interfaces. *International Journal of Human-Computer Studies*, 69(12):854–869.

Kosinski, R. J. (2008). A Literature Review on Reaction Time. (September).

Kovatsch, M., Lanter, M., and Shelby, Z. (2014). Californium: Scalable cloud services for the internet of things with coap. In *Internet of Things (IOT), 2014 International Conference on the*, pages 1–6. IEEE.

Krumm, J. (2012). How People Use Their Vehicles: Statistics from the 2009 National Household Travel Survey. *SAE 2012 World Congress & Exhibition*, pages 1–12.

Lancaster, R. and Ward, R. (2002). The contribution of individual factors to driving behaviour: Implications for managing work-related road safety. Technical report.

Laukkonen, J. (2013). What is a Fuel Gauge?

Laukkonen, J. (2015). A Brief History of the Car Radio.

M., H., Ridder, S., and Janssen, W. (2002). Review of European Human Factors Research on Adaptive Interface Technologies for Automobiles. *TNO Report*.

McKnight, A. J. and Adams, B. B. (1970). Driver education task analysis. volume ii: Task analysis methods. final report.

Merriam-Webster (2004). *Merriam-Webster's collegiate dictionary*. Merriam-Webster.

Michon, J. (1985). Human Behavior and Traffic Safety. *Human behavior and traffic safety*, pages 485–520.

Michon, J. A. and Smiley, A. (1993). GIDS : Generic Intelligent Driver Support. In Michon, J. A., editor, *Generic Intelligent Driver Support: A comprehensive report on GIDS*, pages 3–18. London and Washington DC: Taylor & Francis.

Miller, S. (2001). Workload measures. *National Advanced Driving Simulator. Iowa City, United States.*

National Center for Statistics and Analysis (2015). Distracted driving 2014 (Traffic Safety Facts Research Note. Report No. DOT HS 812 260). Technical Report April 2016, National Highway Traffic Safety Administration, Washington, DC.

National Center for Statistics and Analysis (2016). Traffic Safety Facts. Report No. DOT HS 812 293. Technical Report June, National Highway Traffic Safety Administration, Washington, DC.

Ng, A. W. Y. and Chan, A. H. S. (2012). Finger Response Times to Visual , Auditory and Tactile Modality Stimuli. II.

NHTSA (2012). Visual-manual NHTSA driver distraction guidelines for in-vehicle electronic devices. Technical report, National Highway Traffic Safety Administration.

NHTSA (2014). TRAFFIC SAFETY FACTS 2014. Technical report.

Nielsen, J. (1993). Usability Heuristics. In *Usability Engineering*, pages 115–163. Elsevier {BV}.

Nielsen, J. and Landauer, T. K. (1993). A Mathematical Model of the Finding of Usability Problems. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, CHI '93, pages 206–213, New York, NY, USA. ACM.

Nilsson, G. (2004). Traffic Safety Dimensions and the Power Model to Describe the Effect of Speed on Safety. *Velocidade.Prp.Pt*.

Novák, V., Perfilieva, I., Republic, C., and Logic, F. (2001). The Principles of Fuzzy Logic: Its Mathematical and Computational Aspects. In Di Nola, A. and Gerla, G., editors, *Lectures on Soft Computing and Fuzzy Logic*, pages 189–237. Physica-Verlag HD, Heidelberg.

OpenStreetMap Wiki (2014). Main Page — OpenStreetMap Wiki.

Osgi (2011). OSGi Service Platform Core Specification The OSGi Alliance. *Cycle*, (April):1–358.

Östlund, J., Nilsson, L., Carsten, O., Merat, N., Jamson, H., Jamson, S., Mouta, S., Carvalhais, J., Santos, J., Anttila, V., Sandberg, H., Luoma, J., de Waard, D., Brookhuis, K., Johansson, E., Engström, J., Victor, T., Harbluk, J., Janssen, W., and Brouwer, R. (2004). HASTE Deliverable 2 – HMI and Safety-Related Driver Performance. Technical report, European Commission Competitive and Sustainable Growth Programme.

Perchonok, K. and Pollack, L. (1981). *Luminous requirements for traffic signs.* [Washington, D.C.:] U.S. Gov. Print. Off.

Pérez, M. E. G., Pedra, M. P., and Font-Mayolas, S. (2008). La distracción de los conductores: un riesgo no percibido. Technical report.

QUINLAN, J. R. (1986). Induction of decision trees. *Machine learning*, 1(1):81–106.

Quinlan, J. R. (2014). *C4. 5: programs for machine learning.* Elsevier.

R Development Core Team (2008). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria.

Recarte, M. A., Perez, E., Conchillo, A., and Nunes, L. M. (2008). Mental workload and visual impairment: differences between pupil, blink, and subjective rating. *The Spanish journal of psychology*, 11(2):374–385.

Riedmiller, M. and Braun, H. (1992). RPROP - A Fast Adaptive Learning Algorithm. Technical report, Proc. of ISCIS VII), Universitat.

Riesel, D. (2007). A brief Introduction on Neural Networks. *Springer-Verlag, Berlin*, page Second edition.

Risk, I. C., Bosser, T., and Groeger, J. A. (1994). GENERIC EVALUATION METHODOLOGY FOR INTEGRATED DRIVER SUPPORT APPLICATIONS. In *Towards an intelligent transport system. Proceedings of the first world congress on applications of transport telematics and intelligent vehicle-highway systems, November 30-3rd December 1994, Paris. Volume 4.*

Road, T. (1992). *Young drivers.*, volume 53.

Rojas, R. (1996). Neural networks. *Springer-Verlag, Berlin*, page Second edition.

Safety, R. and Index, P. (2016). RANKING EU PROGRESS 10 th Road Safety Performance Index Report. Technical Report June.

Schlegel, R. E. (1993). Driver Mental Workload. In Peacock, B. and Waldemar, K., editors, *Automotive ergonomics*, pages 359–382. London : Taylor & Francis.

Shafer, J., Agrawal, R. R., and Mehta, M. (1996). SPRINT: A scalable parallel classi er for data mining. In *Proc. 1996 Int. Conf. Very Large Data Bases*, pages 544–555. Citeseer.

Shelby, Z., Sensinode, Hartke, K. ., and Bormann, C. (2013). Constrained Application Protocol (CoAP) draft-ietf-core-coap-18. pages 1–118. CoRE Working Group IEEE.

Shinar, D., Meir, M., Ben-shoham, I., and Gurion, B. (1998). How Automatic Is Manual Gear Shifting ? *Human Factors and Ergonomics Society*, 40(4):647–654.

Shneiderman, B. (2010). *Designing the user interface : strategies for effective human-computer interaction.* Addison-Wesley, Boston.

Sofia, M. and Pereira, D. O. (2009). In-Vehicle Information Systems-related multiple task performance and driver behaviour : Comparison between different age groups.

Specht, D. F. (1990). Probabilistic neural networks. *Neural Networks*, 3(1):109–118.

Stanton, N. A., Walker, G. H., and Salmon, P. M. (2015). *Human factors in automotive engineering and technology.* Ashgate Publishing, Ltd.

Swanson, B.-M. (2010). Distracted Driving causes nearly 40%. *automotive fleet*, (2010):20–21.

Taylor, M., Lynam, D., and Baruya, A. (2000). The effect of Speed on the accident frequency on road accident. *Transportation Research Laboratory*, page 56.

Teh, E., Jamson, S., Carsten, O., and Hamish, J. (2014). Temporal fluctuations in driving demand: The effect of traffic complexity on subjective measures of workload and driving performance. *Transportation Research Part F: Traffic Psychology and Behaviour*, 22:207–217.

Tilkov, S. (2007). A Brief Introduction to REST.

Tin Kam Ho (1995). Random decision forests. *Proceedings of 3rd International Conference on Document Analysis and Recognition*, 1:278–282.

UPV (2012). Bus CAN. Automoción.

Values, M., Filter, L. V., and Correlation, H. (2014). Seven Techniques for Dimensionality Reduction. pages 1–21.

Vlakveld, W. P. (2004). New policy proposals for novice drivers in the Netherlands. In *Proc. Behavioural Research in Road Safety: Fourteenth Seminar*, pages 194–204.

Walker, G. H., Stanton, N. A., and Young, M. S. (2001). Hierarchical task analysis of driving: a new research tool. *Contemporary ergonomics*, pages 435–440.

Wickens, C. D. (1984). Processing Resources in Attention, Dual Task Performance and Workload Assessment.

Yoo, R. M., Lee, H., Chow, K., and Lee, H.-H. (2006). Constructing a Non-Linear Model with Neural Networks for Workload Characterization. *Workload Characterization, 2006 IEEE International Symposium on*, pages 150–159.

Zhang, Y. and Owechko, Y. (2004). Driver Cognitive Workload Estimation : A Data-driven Perspective. pages 642–647.