DETECTING DRIVER'S FATIGUE, DISTRACTION AND ACTIVITY USING A NON-INTRUSIVE AI-BASED MONITORING SYSTEM

Miguel Costa*, Daniel Oliveira, Sandro Pinto and Adriano Tavares

*Centro Algoritmi, Universidade do Minho,
4800-058 Guimarães, Portugal*
*E-mail: miguelcosta_mapc@live.com.pt

Abstract

The lack of attention during the driving task is considered as a major risk factor for fatal road accidents around the world. Despite the ever-growing trend for autonomous driving which promises to bring greater road-safety benefits, the fact is today's vehicles still only feature partial and conditional automation, demanding frequent driver action. Moreover, the monotony of such a scenario may induce fatigue or distraction, reducing driver awareness and impairing the regain of the vehicle's control. To address this challenge, we introduce a non-intrusive system to monitor the driver in terms of fatigue, distraction, and activity. The proposed system explores state-of-the-art sensors, as well as machine learning algorithms for data extraction and modeling. In the domain of fatigue supervision, we propose a feature set that considers the vehicle's automation level. In terms of distraction assessment, the contributions concern (i) a holistic system that covers the full range of driver distraction types and (ii) a monitoring unit that predicts the driver activity causing the faulty behavior. By comparing the performance of Support Vector Machines against Decision Trees, conducted experiments indicated that our system can predict the driver's state with an accuracy ranging from 89% to 93%.

Keywords: driver monitoring system, intelligent transportation systems, driver distraction monitoring, driver fatigue monitoring

1 Introduction

Motor vehicle traffic crashes are among the leading causes of death all over the world, being driver inattention the most relevant source of the reported injuries [1]. A study published by the National Highway Traffic Safety Administration (NHTSA), in 2015, reported that 94% of traffic accidents are caused by driver-induced errors. Specifically, distraction and fatigue are the most frequent causes, representing 41% of the injuries [2]. In the years to come, addressing these issues will be critical to the breakthrough of the autonomous driving paradigm. This advent will have major socio-economic implications on society, reshaping its lifestyle and redefining its key-infrastructures. Fueled by these concerns and challenges, the academia and industry are pushing the technology boundaries to achieve more practical and viable solutions. As such, Driver Monitoring Systems (DMSs) capable of (i) supervising fatigue and distraction, and consequently (ii) avoiding dangerous driver behavior, have received great attention from the academia and automotive industry [3, 4].

Moving towards a society of automated driving environments raises a set of new challenges in the context of DMSs [5]. In such scenarios, the vehicle performs almost all tasks required for the driving operation. However, in case of an emergency, an automation level equivalent to Society of Automotive Engineers (SAE) 2 or 3 requires the driver to be ready to resume control within the shortest time possible, as the vehicle gets out of its Operational Design Domain (ODD) - the specific conditions under which a given automation system is designed to function [6]. Due to the lack of active involvement in the driving task, drivers may feel a natural tendency to engage in non-driving related activities, which reduces their alertness and, consequently, the ability to perform the fallback operation [7] - the response given by the driver to achieve a minimal risk condition as the automation system leaves its ODD [6]. Some studies proved that the quality of the fallback process decreases as the complexity of driving non-related tasks increase [8, 9]. This complexity is intrinsically dependent on the type(s) of distraction unleashed: (i) visual, (ii) manual, or (iii) cognitive. Furthermore, other studies [10, 11] also show that drivers in a partial or conditional autonomous scenario may become fatigued and drowsy faster than on manual driving scenarios. Clearly, a fatigued or distracted driver has reduced situation awareness, which could lead to delayed reactions and impaired driving performance in a fallback situation [10, 11]. We envision that future in-vehicle information systems will have the ability to adapt the cockpit according to these type of driving situations, counteracting dangerous driver behavior; however, this requires techniques for the real-time monitoring of the driver's state [12].

To overcome this problem, automotive key players have proposed different driver monitoring mechanisms, exploring new sensing technologies and software algorithms. Traditional physiological sensors, such as those capable of acquiring electrocardiograms or electroencephalograms have demonstrated an impressive capacity to assess the driver state in terms of distraction, stress, and fatigue. Nonetheless, due to their intrusive nature, they are only used for validation purposes [3, 13]. In fact, the current state-of-the-art monitoring systems are based on non-intrusive sensors, embedded in the vehicle's cockpit, designed to acquire either (i) driver's performance data, such as vehicle speed and steering movements, or (ii) driver's facial and body pose features, such as eye gaze and head pose [3, 14, 15]. A notable advantage of the later monitoring systems over the former ones is their compatibility with partial and conditional autonomous vehicles. Since in such scenarios the driving task is performed by the vehicle itself, the driver's performance cannot be assessed, and consequently, the systems are not able to infer the driver's state, which is preponderant in a fallback situation.

Despite not being as accurate as traditional physiological sensors, facial and body pose-based systems present an excellent trade-off between fatigue and distraction detection, which can be improved using data analytic techniques. It is clear that Artificial Intelligence (AI) areas, such as Machine Learning (ML), play a significant role in the context of advanced driver assistance and monitoring systems [16, 17]. Empowered with such emerging challenges, we present in this paper a DMS capable of supervising the driver in terms of fatigue, distraction, and activity, by exploiting non-intrusive sensor data and features, assisted by ML models - Support Vector Machines (SVMs) and Decision Trees (DTs). Concerning the fatigue assessment, the system was designed to issue a warning as soon as the driver shows some signs of drowsiness. In terms of distraction, we covered the whole range of distraction types, the driver might be exposed to [3, 18]: (i) visual, (ii) manual, and (iii) cognitive. The classification of the distraction state is enhanced by an activity assessor that recognizes four out of the ten deadliest driving non-related activities [19]: (i) using a mobile phone, (ii) looking to an external event, (iii) interacting with the infotainment, and (iv) interacting with passengers. As a matter of fact, current DMSs does not provide such a comprehensive classification of the driver's state. In particular, frameworks to supervise driver distraction focus only on a specific type of distraction and cannot infer which driving non-related activity is causing the faulty behavior. In this paper, however, we follow a holistic approach, presenting a novel architecture for DMSs. Conducted driving simulator experiments revealed that our system predicts the driver's state with an average accuracy between 89% to 93%.

2 Background

May and Baldwin [20] defined two types of driver fatigue: (i) sleep-related and (ii) task-related fatigue. The former results from sleep deprivation, extended duration of wakefulness and time-of-day (circadian rhythm effect); the latter is related to mental overload or underload, caused by certain characteristics of the driving context, such as task demand and duration. Therefore, task-related fatigue can be sub-classified as active or passive. Active fatigue is caused by high demand and high mental workload driving situations, while passive fatigue is caused by predictable and monotonous tasks. By applying this fatigue model to an autonomous driving scenario, it can be concluded that higher automation levels promote passive fatigue and decrease the active component, as the driver's only task is to monitor the system and take over vehicle control when the automation system experiences an ODD departure and requires a fallback maneuver [20, 21].

According to the literature [3, 18], there are three types of distraction which drivers are exposed to: (i) visual, (ii) manual, and (iii) cognitive. The former is induced by tasks that require the driver to look away from the roadway, such as reading or watching a video. Manual distraction occurs when the driver takes away one or both hands from the Steering Wheel (SW) to perform a non-driving related activity, such as eating or manipulating the infotainment system. Cognitive distraction is related to tasks that involve thinking about something other then the driving activity, affecting the driver's mental workload [3, 18]. This category encompasses tasks such as talking to passengers or thinking about something upsetting.

Conventionally, tasks performed by a vehicle's driver are classified as (i) primary, (ii) secondary, or (iii) tertiary tasks [22, 23]. The former consists of real required driving operations to control the lateral and longitudinal movement of the vehicle. In turn, the second category refers to activities that support the primary task and are important for traffic safety, such as activating indicators or headlights. The tertiary task includes convenience activities that are not related to the driving task, such as controlling the infotainment system. This approach can be successfully applied to classify the

drivers' tasks in vehicles with no driving automation or with simple driver assistance systems (SAE level 0 or 1). However, the paradigm shift caused by autonomous driving is disrupting the traditional control of a vehicle by drivers (decision-making and operation), and therefore, the activities performed by them [24]. In fact, in partially automated driving (SAE level 2), the driver's main task is the monitoring of the vehicle's operation within the road environment, while in conditional automated driving (SAE level 3), the driver only needs to perform fallback operations, with no need to monitor the vehicle. In highly or fully automated driving (SAE level 4 or 5), the driver acts as a simple passenger [6]. In this context, actTypes3 classified the diver's tasks into (i) driving-related tasks and (ii) non-driving related tasks. In manual driving, the former comprises all activities required to safely control the vehicle, i.e., the traditional primary driving task, or to increase driving safety and performance, i.e., the former secondary task. In partial and conditional automated driving (SAE levels 2 and 3), it includes activities related to the supervision of driving environment and fallback operation, while in highly and fully automated mode (SAE levels 4 and 5), it only includes simple activities, such as setting the destination. The non-driving related tasks, as expected, refers to tasks that are not related to driving, such as managing infotainment systems or communicating with passengers. This also includes new activities that will become possible with autonomous driving, such as reading, eating or even sleeping. Since this work aims to design and develop a DMS for the manual and conditional automated driving contexts (SAE levels 0 to 3, inclusive), future references to types of driving activities will use the classification proposed by actTypes3.

Typically, non-intrusive DMSs take advantage of the driver's visual behavior to infer his/her state. Consequently, it is essential to review the conventional concepts about eye movements inherent to the humans' physiology [25]:

1. **Saccades**: This term refers to the simultaneous movement of both eyes. It aims to move the fovea, the only eye region capable of perceiving images with high acuity, rapidly from one point of interest to another [25]. The ballistic aspect of this movement allows the driver to focus on various parts of the visual world, gathering in-

formation to build a mental map of the observed scene.

2. **Fixations**: A state of relative stability, where the gaze is held upon a target in the field of view, allowing the visual system to take detailed information about what is being observed [25]. Generally, a fixation takes more than 100 milliseconds, with the duration often used as a metric that reflects the complexity of the visual information the human wants to extract [26].

3. **Blinks**: A rapid closing and reopening movement of the eyelid that produces tears and spreads them evenly across the front surface of the eye, maintaining it smooth and clear. The average blink rate shows high inter-individual variability, ranging from 10 to 30 blinks per minute [27]. This frequency may be affected by environmental factors, physical activity, cognitive workload, and fatigue [25].
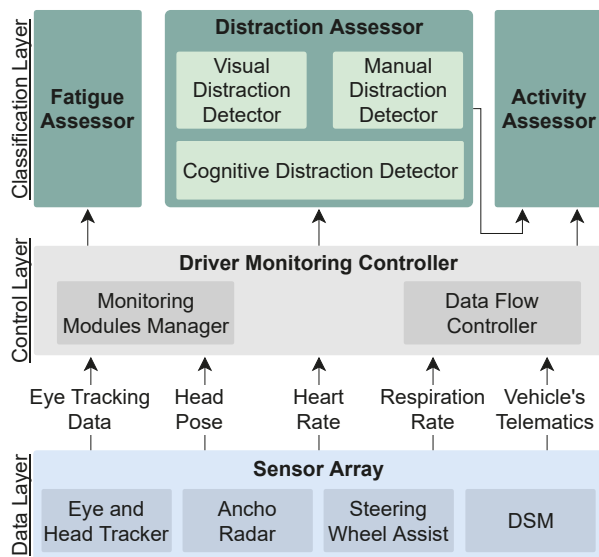
3 DMS Architecture

Figure 1. Overall DMS architecture

Figure 1 depicts the architecture of the proposed DMS, comprising three main layers serving specific tasks: (i) data, (ii) control, and (iii) classification. The former encompasses the sensor array module, which encloses all sensory systems that collect driver-related data. To provide a full monitoring of the driver's state, the overall monitoring system relies on a multi-modal dataset, extracting features related to the driver's physiology and ve-

hicle's telematics. The core unit of the system is integrated into the control layer. This layer is responsible for managing the driver monitoring modules contained in the upper layer and for controlling the data that flows to them. The upper layer (classification) is composed of three main modules and is responsible for classifying the driver's state in terms of fatigue, distraction, and activity.
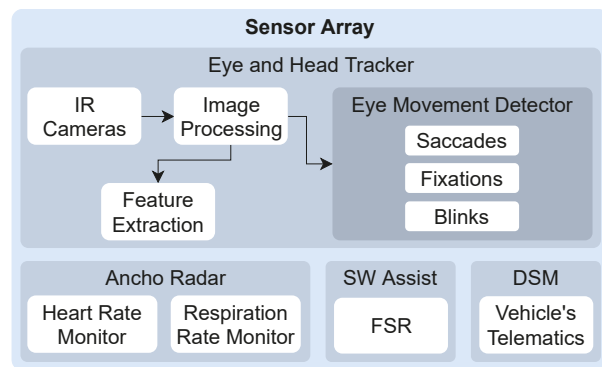
3.1 Data Layer

Figure 2. Sensor array - architecture

The architecture of the sensor array block is detailed in Figure 2. In order to take full advantage of the driver visual behavior, the system relies on a commercial eye and head tracker (Smart Eye Pro (SEP) [28]), with four InfraRed (IR) cameras that can be distributed over the cockpit's dashboard. Since the cameras are only sensible to IR light, it is possible to capture the driver's face under different lighting conditions, even if the driver wears sunglasses or eyeglasses. The software platform employs a set of image processing techniques, which allows to extract a batch of relevant features from the driver's eyes and head, as well as to identify three different eye movements: (i) saccades, (ii) fixations, and (iii) blinks. Regarding the head tracking path, this sensor can estimate the 3D pose of the subject's head, in terms of position and orientation, within a predefined world coordinate system. The eye tracking sub-module, besides detecting the aforementioned eye movements is also responsible for getting features related to eyes' orientation, eyelids' opening, and pupils' diameter. In addition to these functionalities, the SEP platform employs mechanisms that allow uploading a world model, describing the driver's field of view, and infer, in real-time, which scenario's element is being observed. We considered the driver's field of view

divided into four quadrants, two of which divide the road straight ahead. Some additional elements necessary to assist the driver, such as the side and rear mirrors and the cluster were defined in the world model as well. All aforementioned measures are provided along with a quality factor, useful to determine the validity of a sample.

Even in the field of physiological features, we used a commercial contactless biometric sensor (Ancho Radar [29]) to estimate the driver's heart and respiration rate. Such sensor relies on radio frequency waves that measure and capture small movements in the human body. When correctly placed in the cockpit, it can track the driver's chest, which enables the computation of the heart and respiration rate in real-time.

Along with these tracking systems, the sensor array also features a device to detect the presence of both, one, and none of the driver's hands in the vehicle's SW. For this purpose, it uses two Force Sensitive Resistors (FSR), located on each side of the SW, generating two signals that reflect the conductivity of the sensor. Remaining telematics are outputted from a Driving Simulator Mockup (DSM) [30], used as a testing platform for the DMS, which allowed a wide range of data measures to be collected including driving performance metrics (e.g. daytime and vehicle's automation level).

3.2 Control Layer

As aforementioned, the driver monitoring controller acts as the central unit of the entire system. Its functionality is split into two sub-modules: (i) the monitoring modules manager and (ii) the data flow controller (Figure 1). The former is a simple graphical user interface that controls each of the three monitoring units (fatigue, distraction, and activity assessor modules), enabling the user to orthogonally switch on/off any of these units. Every time the driver updates this specification, a message containing the new settings is sent to the data flow controller, which updates the list of signals, yield by the data layer, that are sent to the classification layer. Regardless of the data source, the data flow controller must check the quality parameter of the measures received, forwarding them to the respective monitoring units only when the quality standards are met.
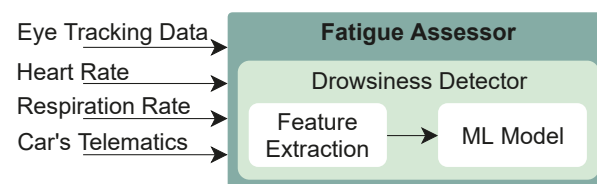
3.3 Fatigue Assessor

Figure 3. Fatigue assessor - architecture

The assessment of driver's fatigue state relies on the classification of physiological and driving context features by an ML model (Figure 3). The physiological features are provided by the SEP and the Ancho Radar. The SEP monitors the driver's eyes, outputting the opening values of both eyelids and the result of the blink detection filter used to calculate the PERcentage of eye CLOsure (PERCLOS) and the blink rate, respectively. The Ancho Radar was specifically introduced to measure the respiration and heart rate of the driver, which proved to be a reliable set of data in fatigue detection. In order to reduce the effect of noisy data points, the measures received from both sensors are submitted to a median filter with a window size set to 3. The described set of physiological features is calculated using a sliding window that grabs the data of the past 20 seconds, with a step size of 1 second. Every time a sliding window is filled up, this monitoring module issues a query to the DSM, which replies with the current time-of-day (day or night) and automation level of the vehicle. After normalizing these 6 features (Figure 4) to the range [0, 1], the system employs a binary ML model to classify the driver's state as attentive or fatigued. The construction of this ML model is described in Section 4.
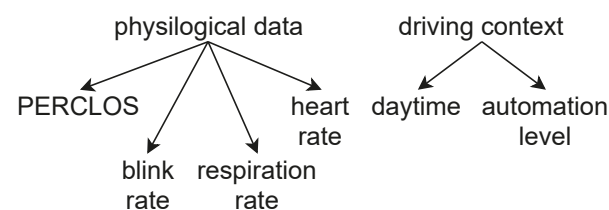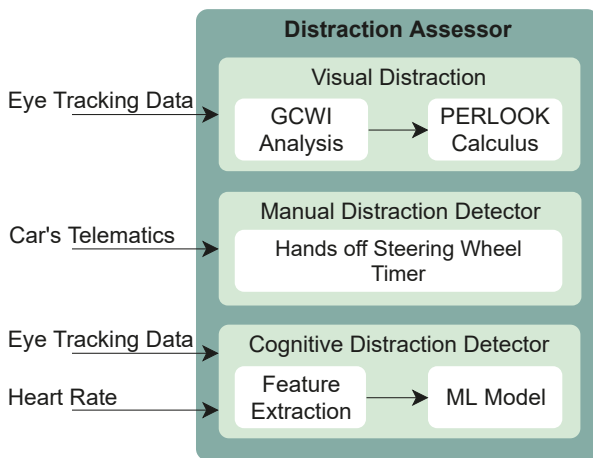

Figure 4. Features for fatigue detection

3.4 Distraction Assessor

The distraction assessor unit (Figure 5) provides a complete monitoring of the driver's distraction state, by offering specialized sub-modules in the de-

tection of visual, manual and cognitive distraction. These sub-modules are executed in parallel and they operate independently of each other.
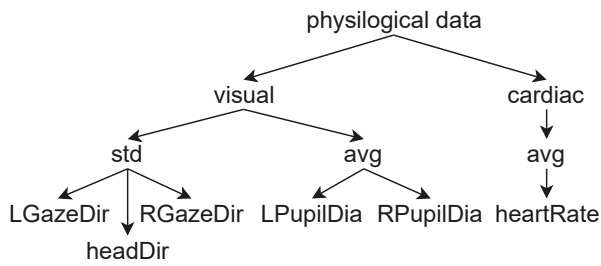

Figure 5. Distraction assessor - architecture

In order to monitor visual distraction, the system takes advantage of the SEP to infer which element of the world model is the driver looking at. From the analysis of the Gaze Closest World Intersection (GCWI), the eye tracking system outputs a variable that names the intersected world element, which, as detailed in Figure 5, is used to compute the PERcentage of time the driver is not LOOKing ahead (PERLOOK). More specifically, this parameter is calculated over a sliding window, which accumulates the name of the gaze intersected elements for the past 10 seconds, with a step size of 1/60 seconds. This step size corresponds, in practice, to the inverse of the SEP sampling frequency (60 Hz), ensuring that every time the window advances only one sample is removed, being also only one added. This configuration matches the one used by RW-VisualDist1, which proposed and evaluated the PERLOOK for visual distraction inference. To calculate the PERLOOK, this module considers that the Field Relevant for Driving (FRD) is only composed by the road straight ahead, which is divided into two quadrants in the SEP's world model. Every time the driver's gaze fades away from the FRD, the system increases the PERLOOK parameter. Exceptions are made when the driver focus the side or rear mirrors, or the cluster. In this case, the PERLOOK is only incremented after 1 second. This *modus operandi* meets the premise defined by the AttenD algorithm [31], which states that eye glances for driving support elements are only considered as

a distraction when they take more than 1 second. Similar to the system proposed by RW-VisualDist1, if the PERLOOK rises above 35%, the driver is regarded as visually distracted and a warning is issued.

For manual distraction assessment, the system is continuously checking the presence of the driver's hands on the SW. For this purpose, the system relies on the SW assist sensor included in the data layer. Every time the driver takes one or both hands away, the manual distraction assessor starts to count the time that passes until the driver puts his/her hands back on. Similar to Mercedes Steer Assist technology [32], if the difference is greater than 10 seconds, the system considers that the driver is potentially distracted.

The monitoring of cognitive distraction is based on the classification of physiological data extracted from the driver's visual behavior and cardiac status. The driver's visual behavior is captured by the SEP platform, while the cardiac status is provided by the heart rate monitor, embedded in the Ancho Radar. The former operates at a sampling frequency of 60 Hz, while the later at 0.5 Hz. Since the second sensor operates at a much lower frequency than the first, an oversampling technique is used to replicate the most recent heart rate for each data packet received from SEP. The received measures are further submitted to a median filter with a window size set to 3, in order to reduce the presence of noise and outliers. From the SEP, the system gets a series of features that enables the calculation of the standard deviation of the left and right gaze direction, as well as the standard deviation of the head orientation and the average driver's pupils' diameter. From the Ancho Radar output, the cognitive distraction assessor calculates the average driver's heart rate. This set of features, summarized in Figure 6, are calculated using a sliding window that grabs the data of the past 5 seconds, with a step size of 1/60 seconds. After being normalized to the range [0, 1], these features are used as input of an ML model, which labels the driver's state as normal or cognitively distracted. The development of this ML model is described in Section 4.

physilogical data

visual → std, avg ; cardiac → avg

std → LGazeDir, headDir, RGazeDir ; avg → LPupilDia, RPupilDia ; avg → heartRate

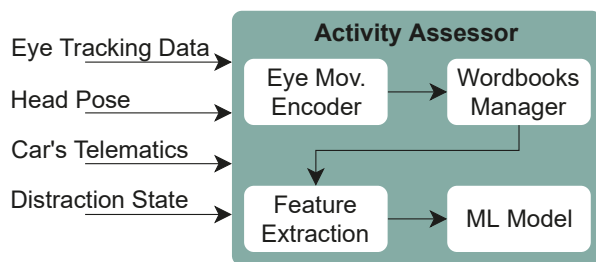**Figure 6**. Features for cognitive distraction detection

3.5 Activity Assessor

This unit is responsible for detecting non-driving related activities that imply, visual, manual, or cognitive distraction. The identified activities, along with the distraction type normally unleashed by each of them [3], are detailed in Table 1.

Table 1. List of detected activities

| Activity	Type(s) of Distraction
Using a mobile phone	Manual, Visual, Cognitive
Looking to an external event	Visual
Interacting w/ the infotainment	Manual, Visual, Cognitive
Interacting w/ passengers	Visual, Cognitive

As depicted in Figure 7, the driver's activity supervision is performed by four sub-modules mutually correlated: (i) eye movement encoder, (ii) wordbooks manager, (iii) feature extraction, and (iv) an ML model.

Activity Assessor

Eye Tracking Data, Head Pose, Car's Telematics, Distraction State → Eye Mov. Encoder → Wordbooks Manager → Feature Extraction → ML Model

Figure 7. Activity assessor - architecture

The Eye Movement Encoder sub-module is responsible for encoding each saccadic movement, detected by the SEP platform, into a character that reflects its spatial disposition. For this purpose, the system starts by searching for saccadic segments in the current sampling window. Once a complete segment is identified, it decomposes the eye movement into its horizontal and vertical components, by calculating the variation of heading and pitch angles, respectively. The horizontal and vertical amplitudes

of each saccade are then stored in two independent buffers, which are then coded in the characters found along the horizontal and vertical axes of the 2D Cartesian Coordinate System (CCS) specified in Figure 8. For this purpose, this sub-module needs to be previously calibrated for each different driver using the system, in order to define the maximum heading (n_h) and pitch (n_p) angles used during the eye movement encoding. This is mandatory to define the different amplitude regions as depicted in the 2D CCS (Figure 8).
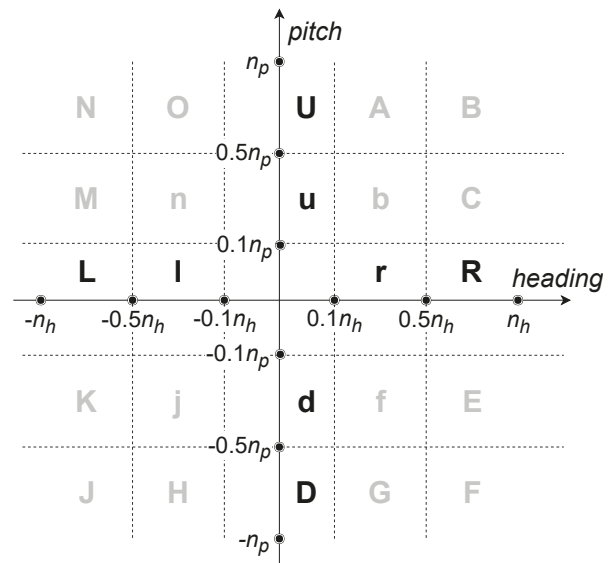
Figure 8. 2D CCS for eye movement encoding

During the encoding process, the heading angle of each saccadic movement is compared against the set of intervals that divide the horizontal axis of this 2D CCS. If it belongs to a small amplitude region ($[-0.5n_h, -0.1n_h[$ and $]0, 1n_h, 0.5n_h]$), the saccadic movement is coded into the characters 'l' or 'r', according to the direction of the movement. However, if it falls in a large amplitude interval ($]-\infty, -0.5n_h[$ and $]0.5n_h, +\infty[$), the movement is coded into the characters 'L' or 'R'. All horizontal saccades that fall into the "dead zone" ($[-0.1n_h, 0.1n_h]$) are not coded, as they are too small to be relevant. The vertical coding is performed in the same fashion, being the pitch angle evaluated along the vertical axis. In order to enable the detection of diagonal eye movements, the returned horizontal and vertical sets of characters are merged as illustrated in Figure 9. This nomenclature is borrowed from the framework proposed by RW-Activity1.
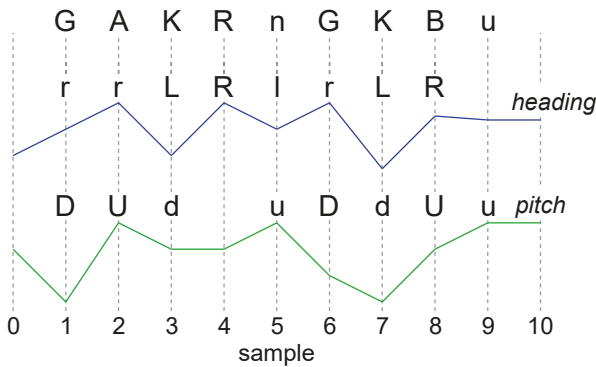
Figure 9. Detection of diagonal saccades

After coding all saccadic sequences in the current sampling window, the system starts assessing the driver's eye movement patterns. In the context of this monitoring application, an eye movement pattern is defined as a string (word) of n successive characters. To accomplish this, the system iterates through the eye movement sequence, successively adding every single movement (character) to four different strings, which encode all movement patterns up to a length of four. Every time a word of length n is built, the system updates the correspondent wordbook. For a pattern (word) that is already registered, its occurrence count is increased. Figure 10 depicts the management of one of the four wordbooks.
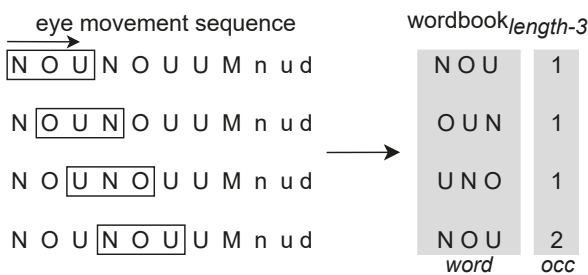
Figure 10. Wordbooks management

The classification of driver activity involves the extraction of an extensive set of 101 features, which can be grouped in different categories. They are calculated using a sampling window of 30 seconds, with a step size of 0.25 seconds. To reduce the presence of noise, all received measures are submitted to a median filter of window size 3. As detailed in Figure 11, from the saccadic eye movement, the feature extraction stage computes 60 statistical features, describing the amplitude and the direction of this eye movement, as well as its distribution over

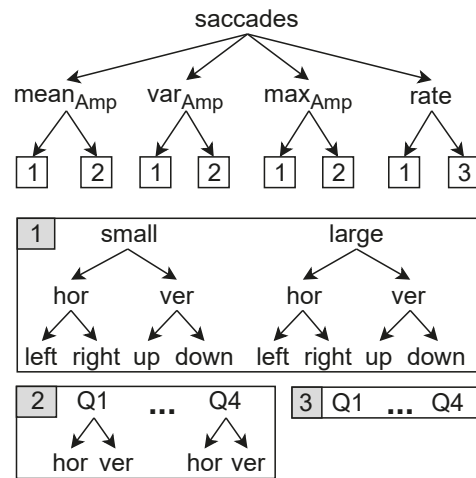the four quadrants that divide the driver's field of view.

Figure 11. Saccadic features

In terms of fixations, the system calculates 3 statistical features concerning the frequency and length of this movement, and another 4 to describe the gaze heading and pitch angles inherent to the visual behavior of the driver. The feature tree showed in Figure 12 provides a deeper view of the set of extracted features.
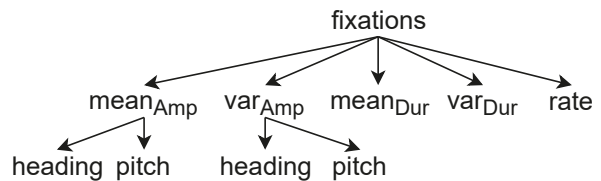
Figure 12. Fixation-based features

Concerning the blink motion, this sub-module computes the mean and variance of blink duration, as well as the blink rate. For each wordbook, it extracts 5 statistical features, describing the occurrence count of the registered words, as well as the wordbook size. More details can be found in Figure 13.

Figure 13. Blink and wordbook features

From the driver's head pose, the system calculates the mean and variance of the head rotation

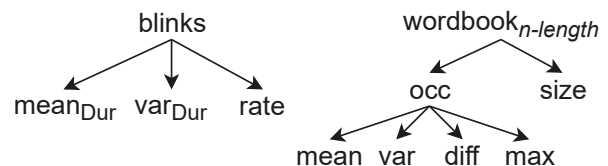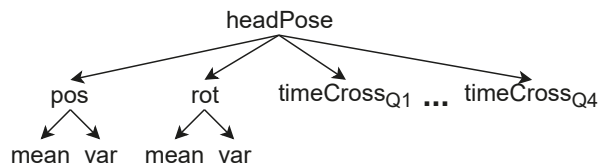and position, while measuring the time at which the head nose vector is directed to each one of the four quadrants dividing the field of view (Figure 14).

headPose
pos   rot   timeCross$_{Q1}$ ... timeCross$_{Q4}$
mean var   mean var

Figure 14. Head pose features

From the vehicle's telematics, the system computes a set of 3 features (Figure 15) that characterizes the driver conduct at the SW. In particular, they reflect the percentage of time driving with one, both and none of the hands on the SW.

handsOnSW
rate
noHands   oneHand   bothHands
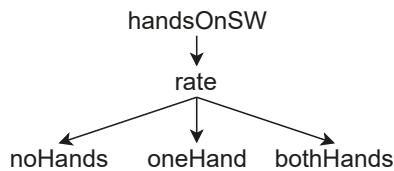
Figure 15. Vehicle's telematics features

The result of the feature extraction stage is further normalized to the range [0, 1], and used as input of a multi-class and supervised ML model, which infers the current driver activity. Such an ML model also takes into account the most recent classification of the driver's distraction state for the final prediction. Therefore, in addition to the aforementioned 101 features, it also takes as input an additional set of 3 binary variables that describe the driver's state in terms of visual, manual, and cognitive distraction.

4 Data Modelling: Methodology

This Section unveils the four stages carried out to build the ML models used to infer the driver's state in terms of fatigue, cognitive distraction, and activity behavior. Manual and visual distraction are predicted by simply comparing single features with heuristic thresholds, while fatigue, cognitive distraction, and activity behavior by using an ML model. For each one of these ML-based monitoring units, we compared the performance of two ML algorithms - SVM and DT, and we then select the one which has the best performance. The results are presented and discussed in Section 5.

4.1 Stage One: Data Collection

A set of experiments were carried out in three different scenarios, designed to collect data to train and test the proposed fatigue, activity, and cognitive distraction assessors. Both scenarios run under the controlled environment of a DSM, equipped with three video-projectors, responsible for reproducing the driving scenario on a curved screen, strategically placed in front of a stationary cockpit.

Regarding the experiment itself, 20 volunteers were recruited to drive along with the three scenarios described in the three next subsections. Each volunteer must meet the following conditions: (i) hold a driver license for more than a year; (ii) drive frequently; (iii) has no major medical illness, including epilepsy; (iv) and has no recent surgery.

4.1.1 Fatigue Assessment

To get a reliable ML model for fatigue inference, we collected data in a scenario mixing highway and city routes. The experiment was divided into two distinct sessions, each one lasting for an hour. During the first session, each one of the 20 subjects drove without sleep deprivation. For this purpose, data were collected during the daytime, ensuring that each subject was submitted to the experiment in the next 10 hours after getting up. The last session was carried out during the nighttime, after the subjects being awakened for, at least, 15 hours. In this phase, all drivers manifested slightly to severe signs of sleepiness. During the driving task, the drivers were asked, every 5 minutes, to classify their attentiveness level according to the Karolinska Sleepiness Scale (KSS): a 9-point Likert scale that measures the subjective level of sleepiness [33]. For data labeling purposes, it was divided into two broad classes: "Attentive" and "Fatigued". The former concerns the KSS levels in the range of 1-5, while the latter refers to the interval of 6-9.

Particularly to this test scenario, the recruited drivers, besides having to meet the requirements exposed in Section 4.1, had to respect (i) a regular sleep schedule, (ii) no alcohol and drug abuse, and (iii) no consumption of medication that disturbs the sleep quality and circadian rhythm. A week before the experiment, they were instructed to go to bed between 23:00 and 0:00 and get up between 7:00 and 8:00. Drugs, alcohol, and caffeine were forbid-

den in the 24 hours before the experiment. Likewise, the use of these substances and napping during the break of the experiment's sessions were also prohibited.

4.1.2 Cognitive Distraction Assessment

The cognitive distraction assessor should be trained and tested in a scenario that does not require frequent maneuvering. It is believed that this situation, very frequent in the city or city-like scenarios, does not allow severe cognitive impairment [34]. In this context, the experiment was conducted in a highway scenario, starting in a low-traffic environment, which increases till a medium level as the driver advances.

The recruited drivers were instructed to follow a leading vehicle, while paying attention to the speedometer to control their speed at 90 km/h. In order to induce cognitive distraction, the driver was subjected to two cognitive tests: (i) one that stimulates the subject's short-term memory and (ii) another that imposes arithmetic loads. The former consists of a *n-back* task, where the driver, in the presence of a stimuli sequence needs to indicate when the current stimulus matches the one from *n* steps earlier in the sequence [35]. More specifically, it was applied a *2-back task*, being the stimulus a series of cardinal numbers between 0 and 9. The second test involves verbally subtracting 7 from 1000, successively.

The experience is divided into four segments: (i) two manual driving sessions, each one lasting 1 minute, with no cognitive distraction induced, and (ii) two driving sessions, each one lasting 2 minutes, with cognitive loads induced by the tests defined above. The attentive driving segments take place at the beginning of the experiment and between the two cognitive tasks.

4.1.3 Activity Assessment

Monotonous roads, such as highways, has proven to be the routes where drivers are more prone to practice distracting activities [36]. Consequently, training and testing the activity assessor with data collected under such route type is essential to get an accurate DMS. We designed a test scenario involving a highway with low traffic, during a sunny daytime simulated environment. In order to collect data

for the set of four distracting activities to be monitored, it was fundamental to divide the scenario into two loops, connected by a conditional link. In the first loop, the road is surrounded by non-distracting environment elements, creating the necessary conditions to collect data on the following three activities: (i) using a mobile phone, (ii) interacting with the infotainment, and (iii) interacting with passengers. The second loop presents a route surrounded by some distracting elements that appear from time to time. More specifically, it was defined that a very uncommon and distracting element might appear every 800 meters. This loop was strictly designed to collect data in situations where the driver is looking to an external event, which constitutes the fourth distracting activity that the system was designed to monitor. The transition between the first and the second loop occurs when the driver turns on the headlights. This transition is not instantaneous, occurring only at the time the driver reaches the end of the first loop.

The driver experience starts with a 30 seconds session of manual driving with no distraction imposed on the driver. After that, the driver is encouraged to sequentially perform the first set of distracting activities, each one lasting 2 minutes. Between two activities there must be a non-distracting driving session of, at least, 30 seconds. After finishing the first set, the participant must turn on the headlights in order to enter in the second loop. Considering that the driver respects the speed limit, this transition can take a maximum of 5 minutes. In the second loop, the driver will be most of the time in normal driving mode, only changing his/her focus when unexpected external elements appear. This loop finishes after the 10th distracting element. At a speed ranging from 60 km/h to 100 km/h, the experiment should take between 17.5 minutes and 23.5 minutes per participant.

4.2 Stage Two: Data Preprocessing

In order to get accurate and unbiased ML models, it is fundamental to train them with balanced and normalized datasets. To balance the collected datasets, we applied the NearMiss-3 algorithm from the NearMiss family of methods [37]. It is an informed undersampling algorithm, which performs undersampling in the majority class(es) based on the distance of its (their) points to the minority class

[37]. When correctly configured, it allows minimizing the number of instances of the majority class(es) among the samples of the minority one. This ensures the reduction of data associated with transitions between classes, and consequently, between states of the driver. By reducing the transition data points, the NearMiss-3 algorithm minimizes the outliers within each label, which is desirable to correctly train an ML model. The data preprocessing stage finishes with the normalization of the resulting datasets to the range [-1, 1].

4.3 Stage Three: Feature Selection

For feature selection, we only considered algorithms of the filter type. They are far more computationally efficient than the traditional wrapper mechanisms, scaling easily to large and high-dimensional datasets. Furthermore, their results are completely independent of the ML algorithm of the classifier [38]. This allows us to train and compare the performance of different ML algorithms without performing the feature selection stage multiple times. From the panoply of available filter methods, we adopted the Correlation Feature Selection, which is based on the premise: "good feature subsets contain features highly correlated with the class, yet uncorrelated with each other" [39]. For this purpose, it adopts a heuristic that evaluates the worth of feature subsets by taking into account the usefulness of individual features for predicting the class label, as well as the level of inter-correlation between them. For each feature subset analyzed, it is given a score in the range [0, 1]. The algorithm returns the feature subset with the highest score.

4.4 Stage Four: Feature Classification

For the feature classification process, we evaluated the performance of two supervised ML algorithms: (i) SVMs and (ii) DTs. SVMs proved to be very efficient in high dimensional spaces, even when the number of dimensions is higher than the number of samples. Despite the expensive computational resources required during the training phase, due to SVM complexity, the algorithm proved to be very memory efficient, as it only uses a subset of training data points (support vectors) in the decision function [40]. This is quite useful in the automotive domain, as it allows the deployment of the monitoring application to low-power platforms. DTs can also

be easily deployed to this type of platforms, as this ML algorithm generates understandable and interpretable rules, which does not require to save any of the training data points - memory efficient. Furthermore, DTs are fast, robust and can handle with high dimensional and large datasets [41]. The steps involved in the training of each one of these ML algorithms are uncovered along this Section.

SVMs are associated with a set of hyper-parameters that must be tuned in order to get the optimal model for the given classification problem [42]. Except for the penalty parameter C of the error term, the set of hyper-parameters that should be tuned varies according to the chosen kernel (Table 2). To tune them, we used a grid search mechanism, implemented by the scikit-learn API [42], which was configured to methodically build and evaluate a model for each combination of hyper-parameters specified in Table 2. Each hyper-parameter was tested with 5 different values, evenly spaced between the respective lower and upper bounds. Each model is evaluated using a 5-fold cross-validation strategy and the model that is selected is the one that presents the best classification accuracy.

Table 2. Hyper-parameters per SVM kernel [42]

Kernel	Function	HP values
Linear	$K(x,y) = xy$	$0.001 \leq C \leq 10$
Polynomial	$K(x,y) = (\gamma(xy) + r)^d$	$0.0001 \leq \gamma \leq 1$
		$0.0001 \leq r \leq 1$
		$2 \leq d \leq 5$
		$0.001 \leq C \leq 10$
RBF	$K(x,y) = exp(-\gamma \|x - y\|^2)$	$0.0001 \leq \gamma \leq 1$
		$0.001 \leq C \leq 10$
Sigmoid	$K(x,y) = tanh(\gamma(xy) + r)$	$0.0001 \leq \gamma \leq 1$
		$0.0001 \leq r \leq 1$
		$0.001 \leq C \leq 10$

When building a DT, it is essential to define the splitting and termination criteria. In the context of the scikit-learn API, this is achieved by simply tuning the hyper-parameters defined in Table 3. For this purpose, we relied on a grid search mechanism, configured to evaluate each built DT according to a 5-fold cross-validation strategy. The hyper-parameters that accept integers or floats as input were tested with 20 different values, evenly spaced between the respective lower and upper limits. Since we did not tune the depth of the DTs to a fixed range, they were expanded until all leaves are pure or not contain the minimum number of

samples to be split, which is defined by the hyper-parameter "min_samples_split".

Table 3. Hyper-parameters of DTs [42]

| HP name	Description	HP values
criterion	Metric for finding the optimal split condition	- gini - entropy
min_samples_split	Minimum number/rate of samples to split a node	[0.05, 1.0]
min_samples_leaf	Minimum number/rate of samples required in a leaf	[0.01, 1.0]
max_features	Maximum number of features considered in a split	$[1, n_{feat}]$

5 Experimental Results

This Section exposes and compares the performance achieved by the SVM and DT algorithms in terms of accuracy, precision, recall, and F1-Score. For a given monitoring unit, we recorded data from 20 participants and the training phase of each algorithm was repeated three times. The SVM and DT models with the best overall statistics are compared during this Section. To perform the driver experiment, the data collection and control modules of the proposed DMS were deployed to a hardware platform running Windows operating system. The labeling of the data was automated by a graphical user interface, embedded in the system's control layer. It provides us with the ability to label the data as it is collected.

5.1 Fatigue Assessment

Figure 16 depicts the Confusion Matrix (CM) describing the performance of the SVM-based fatigue model classifier. It confronts the labels predicted by the model with the ground truth labels of the testing dataset. As observed, the model was able to recognize 88% of the non-fatigue samples and 95% of the samples suggesting signs of fatigue.

From the non-normalized CM, it was calculated the accuracy of the model, as well as the precision, recall, and F1-Score per class (Table 4). This model predicted the driver's state with an accuracy of 91%. The precision metric registered an average of 92%, which implies that 92% of the predictions returned by the model relative to a class are correct. In terms of recall, it was able to detect an average of 91% of the samples reporting a given driver's state. The F1-

Score - a weighted average of precision and recall - assumes an average of 91%.
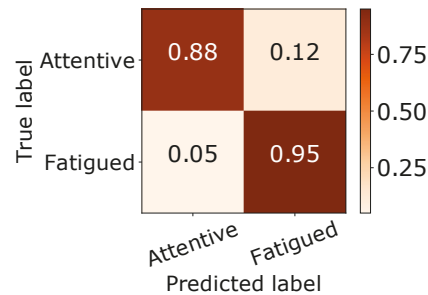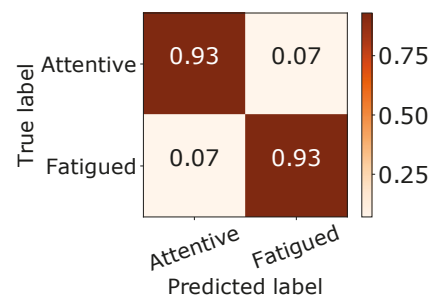

Figure 16. Normalized CM of the SVM fatigue model

Table 4. Classification report of the SVM fatigue model

Class	Precision	Recall	F1-Score
Attentive	0.94	0.88	0.91
Fatigued	0.89	0.95	0.92
average	0.92	0.91	0.91

The CM for the homologous DT model is depicted in Figure 17. As observed, this model registers a drop of 2% in its ability to recognize fatigue situations (true positives). However, the percentage of true negatives increased 5%, meaning this model is less prone to upset the driver with false warnings.

Figure 17. Normalized CM of the DT fatigue model

The classification report for the DT model is provided in Table 5. According to precision, recall, and F1-Score metrics, there is a slight increase in the classification performance, when compared to the homologous SVM model. Furthermore, the DT proved to be more balanced, as the precision and recall, which are complementary metrics, registered the same value. The results also reveal that this model is 2% more accurate. In this context, the

DT was selected to be part of the final fatigue assessor composing the DMS proposed in this paper.

Table 5. Classification report of the DT fatigue model

Class	Precision	Recall	F1-Score
Attentive	0.93	0.93	0.93
Fatigued	0.93	0.93	0.93
average	0.93	0.93	0.93

5.2 Cognitive Distraction Assessment

Figure 18 shows the normalized CM returned by the best-performer SVM model built during the three training trials. As observed, this model was able to correctly predict 92% of the normal driving situations, and 87% of the situations implying cognitive distraction.
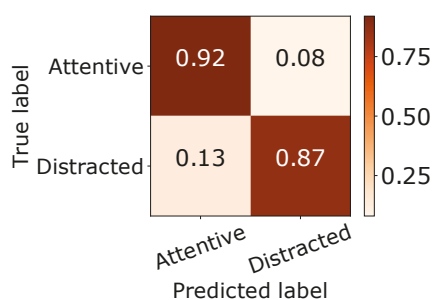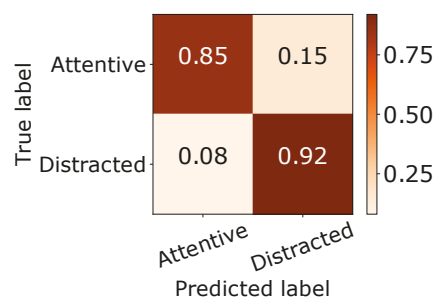

Figure 18. Normalized CM of the SVM cognitive distraction model

The non-normalized CM was used to calculate the accuracy of the model, as well as the precision, recall, and F1-Score per class. The model reached an accuracy of 89%. The values registered by the remaining metrics are exposed in Table 5.8. As observed, the system achieves a very good score in terms of precision (90%) and recall (90%), due to its low false positive and negative rate, respectively. Furthermore, the F1-Score shows a very good balance between these two measures, as it also assumes a value of 90%.

Table 6. Classification report of the SVM cognitive distraction model

Class	Precision	Recall	F1-Score
Attentive	0.87	0.92	0.90
Distracted	0.92	0.87	0.89
average	0.90	0.90	0.90

The CM for the corresponding DT is shown in Figure 19. As it can be observed, this model registers a higher true positive rate (recall) in the recognition of cognitive distraction. In contrast, the recognition of normal driving situations suffered a performance drop of 7%, increasing the number of false positive warnings, which may upset the driver.

**Figure 19**. Normalized CM of the DT cognitive distraction model

In terms of precision, this model exhibits a reverse behavior, being less precise than the homologous SVM in the recognition of cognitive distraction (Table 7). As the F1-Score is a weighted average of precision and recall, we can conclude that the SVM model is more balanced, slightly outperforming the DT. Furthermore, it is also 1% more accurate. Therefore, the SVM model was selected to integrate the cognitive distraction assessor which composes the DMS proposed in this paper.

Table 7. Classification report of the DT cognitive distraction model

Class	Precision	Recall	F1-Score
Attentive	0.91	0.85	0.88
Distracted	0.86	0.92	0.89
average	0.89	0.88	0.88

5.3 Activity Assessment

The normalized CM of the SVM model with the best performance in activity assessment is depicted in Figure 20. As can be observed, the model was able to detect, at least, 90% of the samples reporting a given activity.
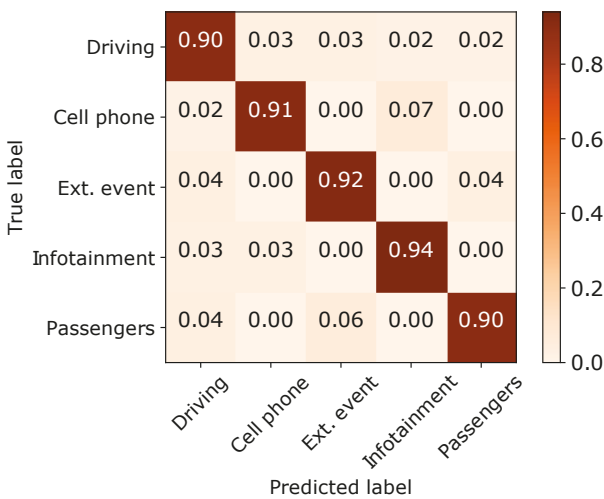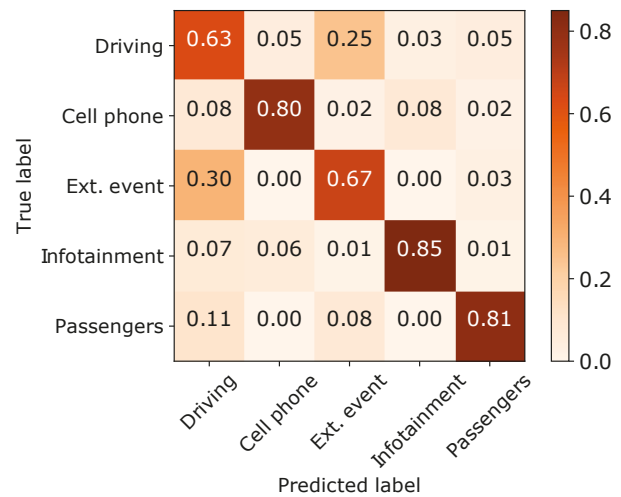
Figure 20. Normalized CM of the SVM activity model

Figure 21. Normalized CM of the DT activity model

To get a deeper insight into the achieved performance, the accuracy of the model, as well as the precision, recall, and F1-Score per class (Table 8) were computed from the non-normalized version of the CM. The results indicated an average accuracy of 91%, which means the model was able to correctly predict the driver's activity in 91% of the predictions performed during the testing phase. The remaining metrics - precision, recall and F1-Score - also registered an average of 91%.

Table 8. Classification report of the SVM activity model

Class	Precision	Recall	F1-Score
Driving	0.88	0.90	0.89
Using a mobile phone	0.94	0.91	0.92
Look. to an ext. event	0.90	0.92	0.91
Inter. w/ the infot.	0.91	0.94	0.83
Inter. w/ passengers	0.94	0.90	0.91
average	0.91	0.91	0.91

Figure 21 shows the normalized CM for the homologous DT. As observed, the true positive rate of all classes is much smaller than those presented by the previous model. This loss is especially evident in the "Driving" and "Looking to an external event" classes, where the true positive rates drop 27% and 25%, respectively. We believe that this loss of performance is an effect of the inability of this algorithm to deal with datasets with many uncorrelated variables, which is a known bottleneck of DTs [41].

In terms of average accuracy, this new model also performs considerably poorer than the previous one, registering a value of 75%. As detailed in Table 9, this behavior is extended to the average value of the remaining metrics, which reveal a performance loss between 20% and 22%. The analysis of the precision, recall, and F1-Score within each class, confirms the already mentioned difficulty of this model to distinguish between the activities "Driving" and "Looking to an external event". In this background, the SVM model was selected to integrate the final activity assessor.

Table 9. Classification report of the DT activity model

Class	Precision	Recall	F1-Score
Driving	0.53	0.63	0.58
Using a mobile phone	0.87	0.80	0.84
Look. to an ext. event	0.66	0.67	0.66
Inter. w/ the infot.	0.88	0.85	0.87
Inter. w/ passengers	0.89	0.81	0.85
average	0.77	0.75	0.76

6   Related Work

RW-Fatigue1 proposed a driver fatigue surveillance system based on the PERCLOS and average eye closure speed parameters. These features were used as input of an Artificial Neural Network (ANN) that classifies the driver's state as fatigued or attentive. The image of the driver's face was acquired using an IR camera, mounted on the vehicle's dashboard, which captured the driver's eyes independently of

the lighting conditions. Following this approach of using gaze-based features, RW-Fatigue2 evaluated the importance of the yawning frequency as an indicator of driver's fatigue. They concluded that merging this new feature with traditional eye closure features - PERCLOS and eye closure interval - by using a robust classifier, such as SVM, provides better results than comparing single features with heuristic thresholds. RW-Fatigue3 introduced a framework to predict driver fatigue situations, using an ANN fed by features derived from head nodding angle, eye gaze and driving time. In terms of head nodding, these authors proved that the head nodding frequency and the mean head nodding angle are very good indicators of the driver's fatigue state. When merged (using an ANN) with time-based indicators, such as time-of-day and time-on-task, and the traditional PERCLOS parameter, the results present an accuracy around 94%. As expected, physiological features such as the heart and respiration rate are also good indicators of the driver's fatigue state [44].

Systems designed to track driver visual distraction are based on the premise that if the driver does not look to the road straight ahead for a certain amount of time, then he/she is visually distracted. Under the lightning of this premise, RW-VisualDist1 proposed a parameter to measure the visual distraction level, the PERLOOK, which reflects the percentage of time spent not looking ahead during a certain time interval. Every time the PERLOOK surpasses 35%, the driver must be warned as the distraction level enters into a dangerous zone. AttenD addressed the research field of driver visual distraction by proposing the AttenD algorithm. The core idea behind their work is that a driver has a time buffer with a maximum capacity of 2 seconds, which is decreased every time the driver drifts his/her gaze away from the FRD. However, if the focus is directed to a driving support agent, such as the side or rear mirrors, or even the speedometer, the buffer will only start to decrease after a time latency of 1 second. When the driver redirects his/her gaze back to the road, the buffer will start to fill up after a period of 0.1 seconds, since the algorithm considers that it took some time until the driver's eyes can focus the road again.

Typical systems that monitor driver's manual distraction rely on capacitive or torque sensors,

which are embedded in the vehicle's SW, being the generated signals evaluated by a control unit. Whenever the amplitude of the signal is insufficient for a predefined time interval, the system regards the driver as manually distracted and issues a warning. This is the approach followed by most of the available commercial solutions, such as IEE's Hands Off Detection [45] and Mercedes Steer Assist [32].

The state-of-the-art monitoring systems to supervise cognitive distraction rely on the analysis of physiological features, mainly derived from the driver's eye gaze and head movement [46, 47]. In fact, it is proved that cognitive loads result in increased gaze concentration within a narrower range [48], which directly affects the driver's visual behavior and head orientation. In addition to gaze and head-based features, RW-CogDist1 also evaluated the relevance of the inter heartbeats interval and pupils' diameter in the recognition of a cognitive distraction situation. They ended up with a system able to recognize the driver's cognitive state with an accuracy of around 93%. RW-CogDist2 proved that merging the traditional gaze and head orientation-based features with driving performance measures, such as steering angle and vehicle's speed can also introduce very promising results.

In the domain of activity recognition, RW-Activity1 pioneered on human's visual behavior. To recognize common office activities, such as reading, watching a video and browsing the web, they recorded the corneo-retinal potential of the human eye, through an electrooculography, and applied a series of algorithms to detect the basic eye movement types - saccades, fixations and blinks - and extract relevant features from them. Since some activities imply repetitive eye movement patterns, these authors proposed a mechanism that maps the amplitude and direction of saccadic movements to a character-based representation. By analyzing the string of returned characters, they were able to identify and explore different eye movement patterns, useful to extract additional statistical features. The processing series of the system terminates with the classification of the extracted data by an ML model. This work was further adapted by RW-Activity2 to a conditionally autonomous driving scenario. The main novelty was the introduction of a head tracking path, capable of measuring the head position

and orientation, and the division of the driver's field of view into quadrants. This last update allowed the system to get an insight where and for how long the driver's eye gaze and head are oriented. RW-Activity2 proved that this novel approach led to an increase in the classification accuracy from 53% to 77%, in the recognition of the following activities: (i) watching a video, (ii) reading, and (iii) writing an e-mail.

6.1 Gap Analysis

This paper combined the findings of previous studies in order to develop a novel system that provides a comprehensive definition of the driver's state, in terms of fatigue, distraction, and activity. For fatigue supervision, we introduced a feature related to the automation level of the vehicle. Regarding the distraction assessment, we ended with a holistic system, capable of providing a full-definition of the driver's distraction state. This classification is even enhanced with the identification of the activity causing the distraction behavior. At the time of accomplishment of this study, the literature only presented standalone systems devoted to a specific distraction type (Table 10). None of them was able to infer the activity causing the faulty behavior. We borrowed the approach proposed by RW-Activity1 and RW-Activity2 for the activity assessment mechanism. We improved their solutions by considering features describing (i) the driver's conduct at the SW and (ii) the type of distraction affecting the driver. This novel approach enabled the recognition of new driving non-related activities. Similar to the approach followed on the fatigue and cognitive distraction assessment mechanisms, we compared the performance of SVM and DT algorithms to classify the set of extracted features.

In the domain of fatigue and distraction supervision, our system does not introduce any improvement in terms of classification performance. However, the same cannot be said in the domain of activity assessment, where our DMS completely outperforms the ones reviewed. Furthermore, to the best of the authors' knowledge, this is the first DMS providing such a complete and holistic classification of the driver's state, completely covering the areas of fatigue, distraction, and activity (Table 10).

7 Conclusion

In this paper, we proposed a non-intrusive, yet accurate, system to provide a full definition of the driver's state in terms of fatigue, distraction, and activity. We proved that it is possible to monitor the driver without relying on the traditional physiological sensors, which may upset and disturb the driving activity itself. In fact, after analyzing the achieved results, we can conclude that facial and body pose features, along with non-intrusive physiological data and vehicle's telematics not intrinsic to manual driving situations, represent the best trade-off between accuracy and ubiquity. In order to exploit the potential of facial features, including the driver's visual behavior, our system relies on a commercial eye and head tracker (SEP), well suited for the demanding environment of a vehicle's cockpit. The largest fraction of the system's dataset is fed by this sensor, like an eye gaze and head pose features dominate in the inference of the driver's state. The remaining physiological data, related to heart and respiration rate, is captured by means of a contactless biometric sensor (Ancho Radar). In terms of vehicle's telematics, our system acquires data describing the presence of driver's hands on the SW, as well as data related to time-of-day and automation level of the vehicle.

For fatigue assessment, our system uses an ML-based approach. In terms of distraction, we designed a holistic system, capable of detecting the three types of distraction that may affect the driver: (i) visual, (ii) manual, and (iii) cognitive. The former two are detected by simply comparing a single variable with a heuristic threshold, while the latter relies on an ML model. Besides detecting the type of distraction affecting the driver, our system is also able to infer which non-driving related activity is causing the faulty behavior. More specifically, we conceived a monitoring unit, which uses an ML model to detect four of the deadliest driving non-related activities: (i) using a mobile phone, (ii) looking to an external event, (iii) interacting with the infotainment, and (iv) interacting with passengers.

For every ML-based monitoring unit, we evaluated the performance of two supervised ML algorithms: (i) SVMs and (ii) DTs. DTs proved to be more efficient in the recognition of fatigue situa-

Table 10. Functionality and performance of previous research on DMSs against our approach

R&D Study/Technology	Fatigue Monitoring	Distraction Monitoring	Activity Monitoring	Performance Acc./Prec./Rec./F1-Score
RW-Fatigue1	Yes	No	No	0.91 / —- / —- / —-
RW-Fatigue2	Yes	No	No	0.95 / —- / —- / —-
RW-Fatigue3	Yes	No	No	0.94 / —- / —- / —-
RW-VisualDist1	Yes	Yes (Vis.)	No	0.98 / —- / —- / —-
AttenD	No	Yes (Vis.)	No	—- / —- / —- / —-
IEE's Hands Off Detection [45]	No	Yes (Man.)	No	—- / —- / —- / —-
Mercedes Steer Assist [32]	No	Yes (Man.)	No	—- / —- / —- / —-
RW-CogDist1	No	Yes (Cog.)	No	0.93 / —- / —- / 0.93
RW-CogDist2	No	Yes (Cog.)	No	0.96 / —- / —- / —-
RW-Activity1	No	No	Yes	—- / 0.76 / 0.71 / —-
RW-Activity2	No	No	Yes	0.77 / 0.70 / 0.76 / —-
Costa et al.	Yes	Yes (Man./Vis./Cog.)	Yes	Fatigue: 0.93 / 0.93 / 0.93 / 0.93 Cog. Dis.: 0.89 / 0.90 / 0.90 / 0.90 Activity: 0.91 / 0.91 / 0.91 / 0.91

tions, reaching an accuracy of 93%. However, they are completely outperformed by SVM models in the recognition of driver's activity, which registered an accuracy of 91%. Concerning the cognitive distraction assessment, this last ML algorithm also proved to deliver more balanced results when compared to DTs, reaching an accuracy of 89%.

8 Acknowledgment

References

[1] World Health Organization, Global status report on road safety 2015, World Health Organization, Tech. Rep., 2015.

[2] S. Singh, Critical Reasons for Crashes Investigated in the National Motor Vehicle Crash Causation Survey, National Highway Traffic Safety Administration, Washington, DC, Tech. Rep., 2015.

[3] C. Craye, A. Rashwan, M. S. Kamel, and F. Karray, A Multi-Modal Driver Fatigue and Distraction Assessment System, International Journal of Intelligent Transportation Systems Research, vol. 14, no. 3, pp. 173–194, Sept. 2016.

[4] G. Turan and S. Gupta, Road Accidents Prevention system using Driver's Drowsiness Detection, International Journal of Advanced Research in Computer Engineering & Technology, vol. 2, no. 11, Nov. 2013.

[5] C. Braunagel, E. Kasneci, W. Stolzmann, and W. Rosenstiel, Driver-Activity Recognition in the Context of Conditionally Autonomous Driving, in 2015 IEEE 18th International Conference on Intelligent Transportation Systems, Sept. 2015, pp. 1652–1657.

[6] SAE International, Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems, SAE International, Tech. Rep., 2016.

[7] M. Cunningham and M. Regan, Autonomous Vehicles: Human Factors Issues and Future Research, in Proceedings of the 2015 Australasian Road Safety Conference, Gold Coast, 2015.

[8] N. Merat, A. H. Jamson, F. C. H. Lai, and O. Carsten, Highly Automated Driving, Secondary Task Performance, and Driver State, Human Factors, vol. 54, no. 5, pp. 762–771, 2012.

[9] J. Radlmayr, C. Gold, L. Lorenz, M. Farid, and K. Bengler, How Traffic Situations and Non-Driving Related Tasks Affect the Take-Over Quality in Highly Automated Driving, Proceedings of the Human Factors and Ergonomics Society Annual Meeting, vol. 58, no. 1, pp. 2063–2067, Oct. 2014.

[10] N. Schömig, V. Hargutt, A. Neukum, I. Petermann-Stock, and I. Othersen, The Interaction Between Highly Automated Driving and the Development of Drowsiness, Procedia Manufacturing, vol. 3, pp. 6652–6659, 2015.

[11] H. Rahman, S. Begum, and M. U. Ahmed, Driver monitoring in the context of autonomous vehicle,

in Frontiers in Artificial Intelligence and Applications, vol. 278. Mälardalen University, Embedded Systems, 2015, pp. 108–117.

[12] A. Amditis, L. Andreone, K. Pagle, G. Markkula, E. Deregibus, M. Romera Rue, F. Bellotti, A. Engelsberg, R. Brouwer, B. Peters, and A. De Gloria, Towards the Automotive HMI of the Future: Overview of the AIDE-Integrated Project Results, IEEE Transactions on Intelligent Transportation Systems, vol. 11, no. 3, pp. 567–578, Sept. 2010.

[13] Q. He, W. Li, X. Fan, and Z. Fei, Evaluation of driver fatigue with multi-indicators based on artificial neural network, IET Intelligent Transport Systems, vol. 10, no. 8, pp. 555–561, 2016.

[14] J. Jo, S. J. Lee, H. G. Jung, K. R. Park, and J. Kim, Vision-based method for detecting driver drowsiness and distraction in driver monitoring system, Optical Engineering, vol. 50, no. 12, pp. 127 202–1–127 202–24, 2011.

[15] T.-H. Chang and Y.-R. Chen, Driver fatigue surveillance via eye detection, in 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), Oct. 2014, pp. 366–371.

[16] O. Gusikhin, N. Rychtyckyj, and D. Filev, Intelligent systems in the automotive industry: applications and trends, Knowledge and Information Systems, vol. 12, no. 2, pp. 147–168, July 2007.

[17] G. Lugano, Virtual assistants and self-driving cars, in 2017 15th International Conference on ITS Telecommunications (ITST), May 2017, pp. 1–5.

[18] S. Maralappanavar, R. Behera, and U. Mudenagudi, Driver's distraction detection based on gaze estimation, in 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Sept. 2016, pp. 2489–2494.

[19] Erie Insurance, Erie Insurance releases police data on top 10 driving distractions involved in fatal car crashes, Erie Insurance Group, Tech. Rep., Apr. 2013.

[20] J. F. May and C. L. Baldwin, Driver fatigue: The importance of identifying causal factors of fatigue when considering detection and countermeasure technologies, Transportation Research Part F: Traffic Psychology and Behaviour, vol. 12, no. 3, pp. 218–224, 2009.

[21] M. Körber, A. Cingel, M. Zimmermann, and K. Bengler, Vigilance Decrement and Passive Fatigue Caused by Monotony in Automated Driving, Procedia Manufacturing, vol. 3, pp. 2403–2409, 2015.

[22] B. Pfleging, M. Rang, and N. Broy, Investigating User Needs for Non-driving-related Activities During Automated Driving, in Proceedings of the 15th International Conference on Mobile and Ubiquitous Multimedia, ser. MUM '16. Rovaniemi, Finland: ACM, 2016, pp. 91–99.

[23] R. Spies, M. Ablaßmeier, H. Bubb, and W. Hamberger, Augmented Interaction and Visualization in the Automotive Domain, in Human-Computer Interaction. Ambient, Ubiquitous and Intelligent Interaction, J. A. Jacko, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 211–220.

[24] B. Pfleging and A. Schmidt, (Non-) Driving-Related Activities in the Car: Defining Driver Activities for Manual and Automated Driving, in Workshop on Experiencing Autonomous Vehicles: Crossing the Boundaries between a Drive and a Ride at CHI '15, 2015.

[25] A. Bulling, J. A. Ward, H. Gellersen, and G. Troster, Eye Movement Analysis for Activity Recognition Using Electrooculography, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 4, pp. 741–753, 2011.

[26] S. Chen and J. Epps, Automatic classification of eye activity for cognitive load measurement with emotion interference, Computer Methods and Programs in Biomedicine, vol. 110, no. 2, pp. 111–124, 2013.

[27] C. Helmchen and H. Rambold, The Eyelid and Its Contribution to Eye Movements, Developments in Ophthalmology, vol. 40, pp. 110–131, 2007.

[28] Smart Eye AB. SE PRO — Smart Eye. (2018, Sept. 29). [Online]. Available: http://smarteye.se/research-instruments/se-pro/

[29] Flat Earth, Inc. Ancho Radar Development Kit - LTSA mini 5-11 GHz – Flat Earth Online Store. (2018, Oct. 9). [Online]. Available: https://store.flatearthinc.com/products/ancho-radar-development-kit-ltsa-mini-5-11-ghz

[30] Ergoneers GmbH, Sim Lab driving simulator — ERGONEERS, (2018, Oct. 29). [Online]. Available: https://www.ergoneers.com/en/hardware/sim-lab-driving-simulator/

[31] K. Kircher and C. Ahlström, Issues related to the driver distraction detection algorithm AttenD, in 1st International Conference on Driver Distraction and Inattention (DDI 2009), Gothenburg, Sweden, Sept. 2009.

[32] Daimler AG, Driver Assistance Systems — Technology Guide, Daimler AG, Stuttgart, Germany, Tech. Rep., 2013.

[33] D. Stawarczyk and A. D'Argembeau, Conjoint Influence of Mind-Wandering and Sleepiness on Task Performance, Journal of Experimental Psychology: Human Perception and Performance, vol. 42, no. 10, pp. 1587–1600, 2016.

[34] R. Nowosielski and L. M. Trick, How Common In-Car Distractions Affect Driving Performance in Simple and Complex Road Environments, in Proceedings of the 9th International Driving Symposium on Human Factors in Driver Assessment, Training, and Vehicle Design. University of Iowa, Nov. 2017, pp. 249–255.

[35] R. Chen, X. Wang, L. Zhang, W. Yi, Y. Ke, H. Qi, F. He, X. Zhao, X. Wang, D. Ming, and P. Zhou, Research on multi-dimensional N-back task induced EEG variations, 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 5163–5166, 2015.

[36] R. Griffin, C. Huisingh, and G. McGwin, Prevalence of and factors associated with distraction among public transit bus drivers, Traffic injury prevention, vol. 15, no. 7, pp. 720–725, 2014.

[37] G. Lemaître, F. Nogueira, and C. K. Aridas, Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning, Journal of Machine Learning Research, vol. 18, no. 1, pp. 559–563, 2017.

[38] C. Huertas and R. Juárez-Ramirez, Filter feature selection performance comparison in high-dimensional data: A theoretical and empirical analysis of most popular algorithms, in 17th International Conference on Information Fusion (FUSION), 2014, pp. 1–8.

[39] M. A. Hall, Feature Selection for Discrete and Numeric Class Machine Learning, 1999.

[40] J. Strickland, Data Analytics Using Open-Source Tools. Lulu.com, 2016.

[41] P. Kashyap, Machine Learning for Decision Makers: Cognitive Computing Fundamentals for Better Decision Making. Apress, 2018.

[42] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[43] M. Sacco and R. A. Farrugia, Driver fatigue monitoring system using Support Vector Machines, in 2012 5th International Symposium on Communications, Control and Signal Processing, May 2012, pp. 1–5.

[44] W. Zhang, B. Cheng, and Y. Lin, Driver drowsiness recognition based on computer vision technology, Tsinghua Science and Technology, vol. 17, no. 3, pp. 354–362, June 2012.

[45] IEE. Hands Off Detection - IEE - a sense for innovation. (2018, Oct. 17). [Online]. Available: https://www.iee.lu/en/products/automotive/hands-off-detection

[46] M. Miyaji, H. Kawanaka, and K. Oguri, Driver's cognitive distraction detection using physiological features by the adaboost, in 2009 12th International IEEE Conference on Intelligent Transportation Systems, Oct. 2009, pp. 90–95.

[47] Y. Liao, S. E. Li, G. Li, W. Wang, B. Cheng, and F. Chen, Detection of driver cognitive distraction: an SVM based real-time algorithm and its comparison study in typical driving scenarios, in 2016 IEEE Intelligent Vehicles Symposium (IV), 2016, pp. 394–399.

[48] J. Engström, E. Johansson, and J. Östlund, Effects of visual and cognitive load in real and simulated motorway driving, Transportation Research Part F: Traffic Psychology and Behaviour, vol. 8, no. 2, pp. 97–120, 2005.

Miguel Costa received his M.Sc. degree in Industrial Electronics and Computers Engineering from University of Minho, Braga, Portugal. His academic education focused on embedded systems and information systems and technologies. During doing his M.Sc. thesis, he addressed the development of a driver monitoring system, acquiring strong background knowledge in the machine learning area and automotive HMI solutions. In the last 2 years, he worked as a Research Fellow at the Embedded Systems Research Group at Centro Algoritmi, hosted in University of Minho. His research interests focus on artificial intelligence and system-on-chip design.

Daniel Oliveira is an R&D Embedded Systems Engineer with a strong background in (real-time) operating systems, embedded security, as well as in automotive HMI systems. He holds an M.Sc. in Electronics and Computer Engineering from University of Minho (Braga, Portugal), and during doing his M.Sc. thesis, he focused on exploiting security technologies provided by Arm processors (Arm

TrustZone). On the last 4 years, he worked as a Research Fellow at the Embedded Systems Research Group at Centro Algoritmi, hosted in the same university. Nowadays, he is a Ph.D. candidate, focused on developing a novel architecture for reconfigurable IoT end-devices, where several constraints, such as the security, performance and power budget must be seriously considered.

Sandro Pinto is a Research Scientist and Invited Assistant Professor at the Embedded Systems Research Group, University of Minho, Portugal. He holds a Ph.D., Masters, and Bachelor in Electronics and Computer Engineering, from the University of Minho. During his Ph.D., Sandro was visiting Ph.D. student at Asian Institute of Technology (Thailand) and University of Wurzburg (Germany). His main research interests include operating systems, virtualization, and security for embedded, cyber-physical, and IoT-based systems. Sandro has authored and co-authored tens of publications, including articles on international journals, conference papers, a book, and book chapters related to embedded and real-time systems.

Adriano Tavares is an Associate Professor at University of Minho, Portugal. He holds a Ph.D. in Industrial Electronics from University of Minho, a Master of Science in Information Technology and an undergraduate degree in Informatics both from University of Coimbra. His research interests are embedded systems modeling and design, system software design, system-on-chip design and engineering education. He is currently teaching embedded and real-time systems modeling and design, compiler design, system-on-chip design, and advanced computer architectures courses at University of Minho. He published more than a 100 book chapters and papers on international conferences and journals related to embedded systems and two books on assembly and C programming.