Alma Mater Studiorum Università di Bologna
Università degli Studi di Modena e Reggio Emilia
Università degli Studi di Ferrara
Università di Parma

MASTER’S DEGREE COURSE IN
ADVANCED AUTOMOTIVE ELECTRONIC
ENGINEERING

Master’s Thesis in Compliance and Design
Of Automotive Systems

Model-Based Validation of Driver Drowsiness Detection System for ADAS

Author:
Ivan Enzo Gargano

University Supervisor:
Prof. Carlo Concari

Company Supervisor:
Ph. D. Roberto Rabbeni

Session III

A.Y. 2020/2021

To my beloved Parents,

To my pillars Nonna Violante, Nonna Emilia and Elena,

To those who have always believed in me,

And were there for me, during hard times.

SUMMARY

« You can't connect the dots looking forward; you can only connect them looking backwards. So you have to trust that the dots will somehow connect in your future. »

Steve Jobs

The work described in this Master's Degree thesis was born after the collaboration with the company *Maserati S.p.a*, an Italian luxury car maker with its headquarters located in Modena, in the heart of the Italian Motor Valley, where I worked as a stagiaire in the *Virtual Engineering* team between September 2021 and February 2022.

This work proposes the validation using real-world ECUs of a Driver Drowsiness Detection (DDD) system prototype based on different detection methods with the goal to overcome input signal losses and system failures. Detection methods of different categories have been chosen from literature and merged with the goal of utilizing the benefits of each of them, overcoming their limitations and limiting as much as possible their degree of intrusiveness to prevent any kind of driving distraction: an image processing-based technique for human physical signals detection as well as methods based on driver-vehicle interaction are used. A Driver-In-the-Loop simulator is used to gather real data on which a Machine Learning-based algorithm will be trained and validated. These data come from the tests that the company conducts in its daily activities so confidential information about the simulator and the drivers will be omitted. Although the impact of the proposed system is not remarkable and there is still work to do in all

its elements, the results indicate the main advantages of the system in terms of robustness against subsystem failures and signal losses.

CONTRIBUTIONS

The main contributions of the work are:

- Partecipation to new ECUs integration on the DIL Simulator.
- Design and implementation of a Python-based prototype system capable of training, testing, and validating Machine Learning classifiers for PERCLOS, Vehicle Lane Position and Steering-Wheel angle.
- In depth analysis of all the aspects for design and implementation of a Driver Drowsiness Detection system from the system engineering point of view.
- Support to driving sessions data logging and to Driver-In-the-Loop simulator enhancement via MATLAB AppDesigner applications and Simulink models.
- Exploratory Data Analysis and data pre- and post-processing.

DISCLAIMER

Due to the confidentiality of some information belonging to this work, there will be several points that will be omitted, in agreement with the academic and company tutors.

OUTLINE

The remainder of this dissertation is organized as follows:

- *Chapter 1* gives information about background and legislative context on which this work is based. Here fundamental concepts, terminology and methods related to drowsiness detection are illustrated to understand the complexity of a DDD system more deeply and why it is needed; furthermore, an overview on the most recent commercial solutions is provided.

- *Chapter 2* describes methodologies and approaches used for this work. Here the core concepts of the Model-Based Design methodology and Machine Learning are exploited to properly state the project goals and specifications.
- *Chapter 3* exposes all the considerations made as a starting point for the system project. The feasibility analysis starts from what can be used for the system project and, after a proper literature analysis, in the second part of the chapter the various driver-based and driving-based indicators extraction are detailed.
- *Chapter 4* focuses on the implementation on top of all the statements and considerations made in the previous chapters. Here there is the description of the system structure, the data fusion methodology and the used Machine Learning algorithm.
- *Chapter 5* is dedicated to the validation procedure of the system and to the exposure of the results.

Finally, some space will be left for the *Future development steps* and *Final thoughts* on the project described in this work.

TABLE OF CONTENTS

LIST OF PICTURES

LIST OF TABLES

1. INTRODUCTION

«Dreams are constructed from the residue of yesterday »

Sigmund Freud

1.1 BACKGROUND AND CONTEXT

Of all the problems related to transport, safety is the one that has a greater impact on the everyday life of citizens. In addition, it significantly affects many socio-economic indicators. A great deal of effort has been made in recent years in the EU on programs of action for road safety, reducing the total number of fatalities: indeed, road traffic accidents in EU decreased by 53% between 2001 and 2016 (Eurostat: Statistics explained, 2019). However, the situation, both in Europe and all over the world, is still far from satisfactory. According the the *World Health Organization (WHO)* report from 2018 on global road safety (World Health Organization, 2018), approximately 1.3 million people die each year because of road traffic crashes, which represents an annual cost for most countries of approximately 3% of their gross domestic product. An *European Commission* white paper from 2011 aimed to halve the number of road deaths by 2020 (European Commission, 2011) but according to its evaluation report (Tsamis, 2021) all these efforts are still not enough to reach this goal, especially by looking at new targets of halving road deaths and serious injuries even more by 2030 (European Transport Safety Council, 2021) or by looking at the *Vision Zero* target imposed by European Commission, - i.e. that there should be no deaths or serious injuries on Europe's roads by 2050 (Tsamis, 2021). *WHO* identifies the inattention of the driver among the main causes of road crashes (World Health Organization, 2018). Additionally, recent studies states that almost a third of fatal and injury crashes involved *driver inattention and distraction* and many of these could have been prevented for instance by deploying system-wide solutions that could mitigate or prevent distraction crashes, or including intervening vehicle safety technologies and infrastructure solutions to promote a forgiving road environment, blocking capabilities within technologies to prevent communications while driving, and interventions communicating the risks associated with inattention (Wundersitz, 2019).

The starting point to deal with this problem is its definition. One common way to define driver inattention is given in (Regan, 2008):

Driver inattention represents diminished attention to activities that are critical for safe driving in the absence of a competing activity.

There are different categories for the driver attention state, but in practice, two main categories are proposed: (1) distraction and (2) fatigue. This work will focus on the fatigue category.

The term *fatigue* refers to a combination of symptoms, such as impaired performance or a subjective feeling of drowsiness and an accepted and sufficiently accurate definition of fatigue is the first step towards achieving the further aims of detecting fatigue and increasing the available means of treating it (Garcia Daza, 2014). Definitions like the previous one, however, have proven elusive, in part because fatigue is a complex phenomenon, involving several psychosocial and behavioral processes (Shen, 2006). As a result, various definitions have been proposed. The *European Transport Safety Council* (*ETSC*) states in (European Transport Safety Council, 2001):

Fatigue is associated to the tiredness concerning the inability or disinclination to continue an activity, generally because the activity has been going on for too long.

There are different kinds of fatigue, such as local physical fatigue (e.g., in a skeletal or ocular muscle), general physical fatigue (following heavy manual labour) or “central nervous” fatigue (sleepiness). The last of these is mental fatigue – not “having the energy” to do anything (Kecklund, 1995).

Besides, *ETSC* establishes *drowsiness* as one external representation of fatigue, being the most important for driving. In this work, the terms drowsiness and fatigue will be employed interchangeably, unless otherwise stated. Therefore, *ETSC* concludes that there are four drowsiness levels, based on user behavior: (1) completely awake, (2) moderately awake, (3) drowsy and (4) severely sleepy. In this work, they will be grouped into two levels, *alert* and *drowsy*, to reduce complexity and to obtain generalized results by using binary classifiers. An example of similar approach is found also in literature, like the one proposed in (Garcia Daza, 2014).

Fatigue during driving has been shown in several studies to result in a greatly increased risk of suffering a collision. In particular, the *U.S. Department of Transportation, National Highway Traffic Safety Administration* (*NHTSA*) showed in (U.S. Department of Transportation, National Highway Traffic Safety Administration, 2006) that that driving with fatigue raises the accident risk between four to six times, compared to alert driving. Furthermore, several research works like (The Royal Society for the Prevention of Accidents, 2001), (Al-Mekhlafi, 2020) and (Abdulbari Bener, 2017) establish a strong relationship between driver fatigue and road accidents and the reason is because the level of vigilance and alertness of the driver under the influence of fatigue would have become worsened enough to undermine driving performance.

Consequently, the development of systems capable of monitoring drivers' fatigue state in real-time and capable of warning them just before they fall asleep is vital to prevent crashes. Those systems are in general referred to the macro-group of *Advanced Driver Distraction Warning* (*ADDW*) systems. *European Commission* has prepared and supported the development of these systems (European commission, 2021). Back to November 2019, the *EU Council of Ministers* passed a general safety regulation mandating automakers to install advanced safety systems in all new cars on the EU market (European Commission, 2019). The new regulations will be gradually implemented over the course of four years, starting in 2022 with all new type-approved cars with a certain level of autonomous driving capability. By 2026, the law will include all newly produced cars on the EU market, no matter their level of automation. With the new regulation in place, it is believed at least 140,000 serious injuries will have been avoided by 2038 (SmartEye, 2020).

1.2 DROWSINESS DETECTION METHODS

Fatigue is hard to measure, because the measuring method is difficult, and its exhibitions are complicated. In recent years, researchers have tried different measuring and analyzing methods to estimate and predict driver's state (Dong, 2009). According to the state-of-the-art, there are five main categories depending on the type of measures used in the detection (Garcia Daza, 2014): (1) driver biological, (2) subjective report, (3) driving performance, (4) driver physical and (5) hybrid measures.

1.2.1 DRIVER BIOLOGICAL SIGNALS

The first category directly measures the biological signals on the driver’s body to properly apply signal post-processing techniques. The most important signals are:

- Electroencephalography (EEG),
- Electrocardiogram (ECG),
- Electrooculography (EOG),
- Surface Electromyogram (sEMG),
- Galvanic Skin Response (GSR),
- Respiration.

The main drawback of such signals is that conventional measuring instruments are intrusive. In addition, most of the biological patterns vary between individuals making the results less general. However, some biological variables, such as ECG, GSR and respiration, can be measured with non-intrusive devices in the vehicle. These include sensors based on seat pressure (Rosario, Solaz, Rodríguez, & Bergasa, 2010), flying instrumented with electrodes (Baek, 2009) and nanosensors (Bal, 2007). These devices are based upon the same technology as conventional devices, but the body contact is achieved indirectly by taking advantage of the driver’s own position while driving. Their main limitation is that the quality of contacts is worse than those made clinically, which means using more sophisticated methods of signal analysis to remove noise and extract the relevant parameters. This causes a performance decrease that makes the proposal impossible in most cases. These signals are normally used in psychological studies and as the ground truth for other non-intrusive methods (Garcia Daza, 2014).

1.2.2 SUBJECTIVE REPORT

There are several subjective measurement scales in use. The two most common are the *Stanford Sleepiness Scale* (*SSS*), proposed by (Hoddes, 1973), and the *Karolinska Sleepiness Scale* (*KSS*), proposed by (Akerstedt, 1990). KSS has become the most employed tool for the subjective self-assessment of drowsiness. It has been shown in several studies to suitably correlate with physiological signs of drowsiness (Kaida, 2006) (Ingre, 2006) .

A supervisor asks the driver from time to time about his or her status while he or she is driving. The scale involves nine steps, of which every number is associated with a label as follows (Doudou, 2020):

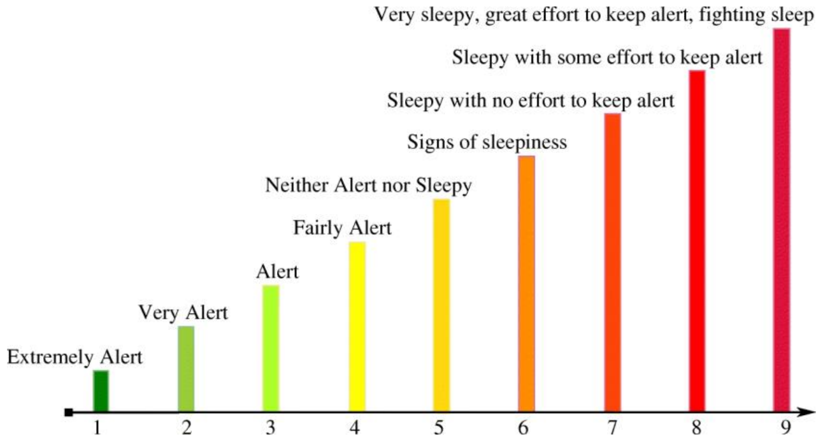

Figure 1-1: Drowsiness scoring using Karolinska Sleepiness Scale (KSS)

A drawback of KSS is its recording over relatively long-time intervals, e.g., every 10-15 min, as a trade-off between high temporal resolution and avoiding intrusive feedback. Therefore, KSS cannot record sudden drowsiness variations caused by different situations. In addition, this kind of approach requires that the driver frequently reports his or her state. Thus, both the drowsiness level result and the driver could cause interference. In brief, this is an intrusive measure that cannot be employed in the development of an automatic monitoring system to warn drivers before they fall asleep at the wheel in real time. As in the previous category, this measure is normally exploited for psychological studies of drowsiness and as the ground truth to test the performance of other proposals (Garcia Daza, 2014).

1.2.3 DRIVING PERFORMANCE

The measured signals on the vehicle reflect the actions of the driver, and therefore, by analyzing them, the driver's behavior can be indirectly characterized in a non-intrusive way. They are usually directly obtained from a simulator or from the internal sensors of a vehicle through the *CAN* (*Controller Area Network*) bus. The force on the pedals, changes in vehicle speed, steering wheel movements, the lateral position or lane changes are typically employed in this category as reported in (Wakita, 2005) and (McCall, 2005).

Indicators of driver drowsiness may be formed by extracting relevant information from signals, such as those mentioned above. These indicators are scalar-valued functions that map given segments of signals onto numerical values. Several indicators, based on individual signals or a combination of them with different complexities have been exploited in literature (Eskandarian A. a., 2007) (Sandberg, 2011).

The advantage of this approach is that the signals have physical meaning, and their acquisition and posterior indicator computation is relatively easy. However, these systems usually require a training period for each person; thus, they are not applicable to occasional drivers. Additionally, they do not work in the detection of "micro-sleeps" – situation where drowsy driver falls asleep for a few seconds on a straight road without changing the direction of the vehicle.

1.2.4 DRIVER PHYSICAL SIGNALS

The physical measures approach is mainly based on monitoring the driver's face, employing cameras and image processing techniques to obtain some physical indicators:

- blink duration,
- blink rate,
- *PERCLOS* (*Percentage of Eye Closure*),
- eye closure duration,
- nodding frequency,

- fixed gaze,
- etc.

The *PERcent of the time a driver's eyelids are CLOSed* (*PERCLOS*) is known as one of the most effective parameters in the drowsiness detection (Bhuiyan, 2009). It measures duration of blinks and eye closures, and proportion of time eyes closed over a specified time interval. This approach is effective, because, on the one hand, driver drowsiness is exposed by the appearance of the driver's face and the activity of his head and eyes. On the other hand, measures can be carried out in a non-intrusive way. However, PERCLOS is normally used in research and simulated scenarios, but not in real ones, due to the problems of vision systems working in outdoor environments (lighting changes, sudden movements, etc.). Additionally, it does not work properly with users wearing glasses and needs high computational requirements.

1.2.5 HYBRID APPROACH

All the previously mentioned methods have strengths and weaknesses:

- Vehicle-based measurements depend on specific driving conditions (such as weather, lighting, etc.) and can be used on specific roads only (with clearly marked signs and lanes). Moreover, they may lead to many false positives (or false negatives), which would lead to a loss of confidence in the method.
- Behavioral measures, on the other hand, may show huge variation in the results depending on the associated lighting conditions.
- Physiological measures are reliable and accurate, but their intrusive nature is still a challenge.

Regarding systems based on a single sensor, they usually have difficulties, due to the uncertainty of the observations associated with the measurements obtained in real conditions. The use of multi-sensor systems reduces the uncertainty and ambiguity of data obtained by a single source. Furthermore, the combination of indicators from different sensors, e.g., combining driver physical with driving performance measures, increases the confidence of drowsiness detection. These proposals are known as *hybrid measurement systems*.

Hybrid systems propose the concept of data fusion. *Data fusion* is the integration of information from different sources (e.g., sensors, vehicle network) into an output that has potentially greater value than the original data from the individual sources (Darrell S. Bowman, 2012).

1.3 COMMERCIAL SOLUTIONS

The automobile industry has spent and will continue to spend a significant number of resources in driver drowsiness detection systems and more in general in driver monitoring systems. Indeed, this market is reported to have a continuous growth due to the joint effort of automotive manufacturing companies – who are working towards developing safety systems to curb the increasing ratio of road fatalities – and governments – which are imposing stringent regulations towads driver and passenger safety and, furthermore, taking various initiatives to reduce road accidents (Automotive Driver Drowsiness Detection System Market Size, Share & Industry Analysis, 2022). This section provides a representative sample of solutions present on the market.

- ***Toyota/Lexus.*** A first example of mass production driver monitoring system is *Toyota's driver monitoring system* (advanced drive specifications), developed by *AISIN* (AISIN, 2021). This system uses a general-purpose

Figure 1-2: Driver monitoring system (camera)

processor to implement artificial intelligence (AI). It has greatly improved the detection of gaze, face direction, eyelid opening/closing. It also has a function to safely stop the car if the driver's driving posture is distorted or the car is unresponsive to warnings (Trends and Future Prospects of the Drowsiness Detection and Estimation Technology, 2021).

- ***Bosch.*** The *Bosch driver drowsiness detection* system monitors steering wheel movements advising drivers to take a break in time. The system is based on the evaluation of 70 signals by an algorithm, which begins recording the driver's steering behavior the moment the trip begins. It then recognizes changes over the course of long trips, and thus also the driver's level of fatigue. Based on the frequency of these changes and other parameters (among them there are for instance the length of the trip, usage of turn signals, and the time of day) the function calculates the driver's level of fatigue. If that level exceeds a certain value, a warning icon flashes on the instrument panel to warn drivers that they need a rest (Bosch, s.d.).
- ***Continental AG***. Continental proposes a modular solution based on cameras, ECUs, and custom algorithm able to perform facial, emotion and gesture recognition. The system monitors the driver's most recent behavior including head position, eye gaze (area and duration) and further signals from the car such as vehicle speed and traffic signs. The driver's drowsiness is diagnosed based on eye blink duration and frequency as well as eye opening and closing velocity. For this, the absolute opening of left and right

Figure 1-3: The Bosch Steering-angle sensor used in their DDD system

eye is constantly being monitored and evaluated. Based on the Karolinska Sleepiness Scale (KSS) an easy to use four level drowsiness scale is calculated ranging from 1 (alert) to 4 (sleepy) (Continental AG, s.d.).

- ***DENSO.*** *DENSO's Driver Status Monitor* uses a camera to capture an image of the driver's face and establishes the driver's condition based on visual analysis. It detects carelessness, distraction, and drowsiness, and then alerts the driver of any potential danger. Because cabin brightness varies signficantly between day and night, DENSO uses near-infrared LED (which is invisible to the human eye) to adjust for darker conditions. This permits clear detection of a driver’s face, variances in facial and physical attributes, and changes in posture. The driver’s facial image is analyzed using *DENSO's Recognition Algorithm* system which identifies facial features, including the eyes, nose, and mouth, as well as facial contours. It detects driver behavior, including eyelid movement and the driver's position, to establish potential dangers, such as drowsiness (DENSO, s.d.).

2 METHODOLOGIES AND APPROACHES

« The most dangerous kind of waste is the waste we do not recognize. »

Shigeo Shingo

This chapter provides an overview of methodologies and approaches used in this work. It starts from *Model-based Design*, a state-of-the-art method in various fields like aerospace, defense, and automotive developments (Bergmann, 2014). After that, the focus will be moved on *Machine Learning*, which is the study of computer algorithms that can improve automatically through experience and using data (Mitchell, 1997).

2.1 MODEL-BASED DESIGN

In a world where engineering organizations, to succeed in competitive marketplace, must adapt to rapid technological change by satisfying a continuous demand for new products and technologies, innovation is not simply a desirable goal: it is a necessity. And it must be paired with their increasing complexity. The pressure

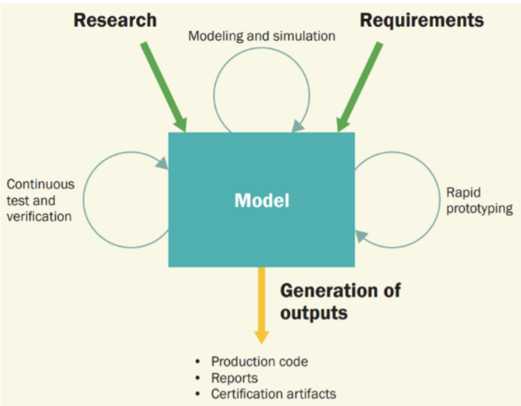

Figure 2-1: Role of the Model in MBD

coming from innovation and complexity for engineering organizations comes from multiple perspectives - not only technical, but also organizational, administrative, and cultural. For this reason, in the past 10 years, organizations have decided to turn towards Model-based Design.

Model-Based Design (*MBD*) is a model-centric approach to the development of control, signal processing, communications, and other dynamic systems. Rather than relying on physical prototypes and textual specifications, Model-Based Design uses a model throughout development. The model includes every component relevant to system behavior - algorithms, control logic, physical components, and intellectual property (IP) (Aarenstrup, 2015). Once the model is developed (elaborated), it becomes the source of many outputs, including reports, C code, and HDL code, even compliant with different standards. Model-Based Design enables system-level and component-level design and simulation, automatic code generation, and continuous test and verification as showed in *Figure 2-1* (Aarenstrup, 2015).

2.1.1 CORE CONCEPTS

MBD is grounded on eight core concepts (Aarenstrup, 2015):

- **Executable specification.** A model which includes all design information, including system requirements, system components, IPs, and test scenarios. In general, it includes more information than a text document and provides less unanbiguous information as well.
- **System-level simulation.** The simulation of the entire system is done to investigate system performance and component interactions. Furthermore, this allows requirements validation, feasibility check and early testing and verification.
- **What-if analysis.** A simulation method which can be used to both test single system components and interactions among multiple system components. This grants the advantage of generating knowledge about the system and to evaluate multiple design ideas.

- **Model elaboration.** An iterative process that uses simulation to turn a low-fidelity system model into a high-fidelity implementation by simulating and refining a starting high-level model. This allows the system to be continuously tested and developed.
- **Virtual prototyping.** A technique that uses simulation to validate a design before hardware is available. Virtual prototypes save development time because building a model is usually much faster than building a physical prototype for experiments on the system.
- **Continuous test and verification.** Practice of simulating a design at every stage of development. It is used to identify faults as soon as they are introduced into the design. This allows to reduce cost and development time. It can take different forms and can be conducted at different levels, depending on the complexity of the system and the stage of development. Some examples might be:
    - *Open-loop testing* - testing a single component with predefined inputs and specified checks for the outputs.
    - *Closed-loop testing* - testing a component or design with a model of the environment and a plant model.
    - *Hardware-in-the-loop* (*HIL*) simulation - generating code from the model of the environment to test an actual embedded system against a simulated environment in real time.
- **Automation.** Using scripts and tools to perform repetitive tasks or tasks that are error-prone when performed manually.
- **Knowledge capture and management.** The models become a common language for the transfer of information within teams and with customers and suppliers. Because the models can be executed or simulated, the knowledge they contain increases as understanding of the system grows.

Figure 2-2 shows a possible MBD application workflow (MathWorks, s.d.).

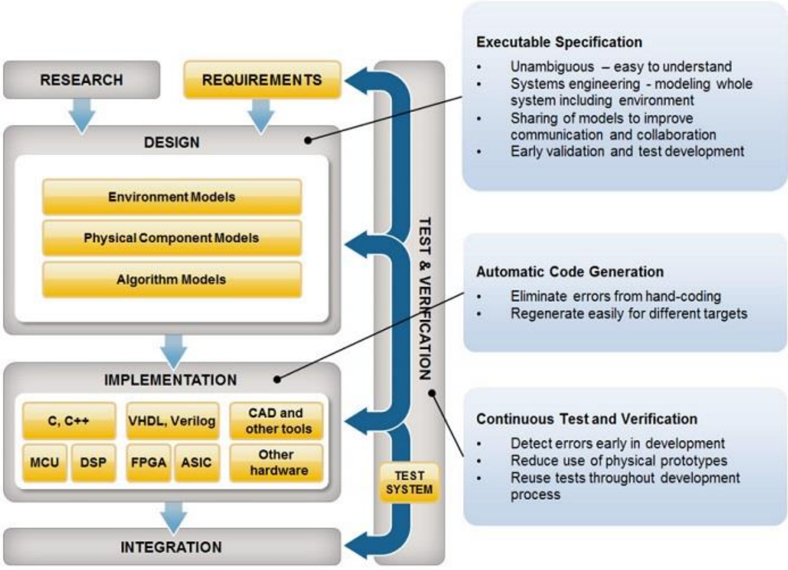

Figure 2-2: Workflow of Model-Based Design

2.1.2 SUPPORT TO COMMON SOFTWARE DEVELOPMENT METHODOLOGIES

The systematic approach that is used in software engineering is sometimes called a software process. A *software process* is a sequence of activities that leads to the production of a software product. Four fundamental activities are common to all software processes (Sommerville, 2015):

- *Software specification*, where customers and engineers define the software that is to be produced and the constraints on its operation.
- *Software development*, where the software is designed and programmed.
- *Software validation*, where the software is checked to ensure that it is what the customer requires.
- *Software evolution*, where the software is modified to reflect changing customer and market requirements.

A simplified representation of a software process is given by the *software process model*, sometimes also called Software Development Life Cycle or SDLC model. This provides an abstract description of software processes that can be used to explain different approaches to software development. In the following session, two of the most important software processes and the support given to them by MBD will be illustrated (Aarenstrup, 2015).

2.1.2.1 WATERFALL MODEL

Waterfall model presents the software development process as several stages, in cascade one with another. Each step must be completed before the next one begins.

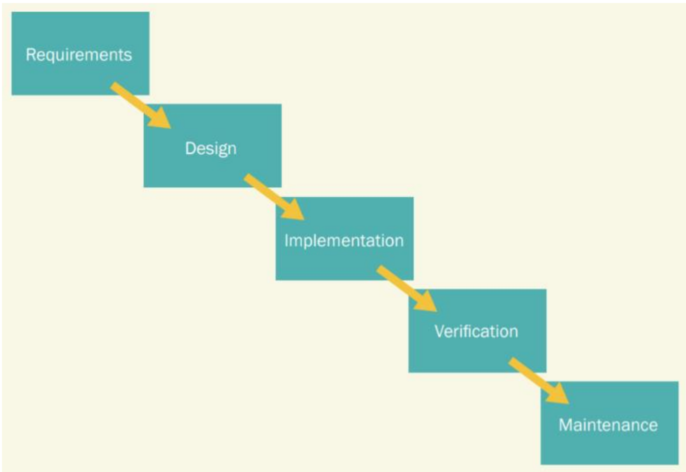

Figure 2-3: Waterfall methodology

In this methodology work and project details are carefully documented to avoid lost of information. One of its drawbacks is its lack of flexibility because the rigid hierarchy is not suitable to handle changes thus not encouraging innovation. Model-Based Design enhances the waterfall methodology in these ways:

- Automation for instance of reports makes easier and less time-consuming repeat tasks.
- System-level simulation makes easier to manage complexity because it allows to show the interactions between system components.
- Easier to try new ideas with What-If analysis.

2.1.2.2 V-MODEL

The V-model represents a plan-driven software development process (also applicable to hardware development) which may be considered an extension of the waterfall model. Instead of moving down in a linear way, the process steps are bent upwards after the coding phase, to form the typical V shape: on the descending side it is possible to find the *Verification phases* and on the ascending one there are the *Validation phases*.

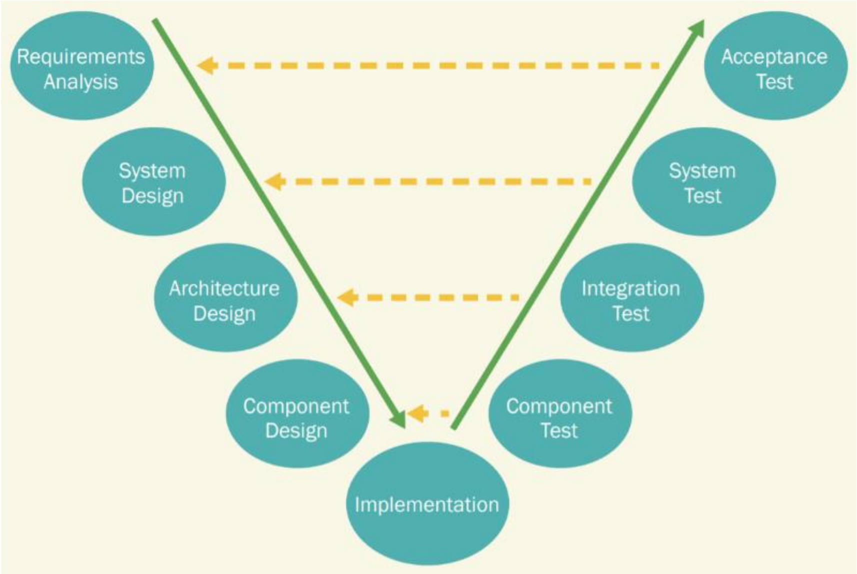

Figure 2-4: The V-model

The drawback of this methodology is the high effort required at the beginning because here is done the entire system development and can be a problem for highly complex systems.

As for Waterfall model, MBD enhances V-model methodology in the following ways:

- Automation for instance of reports makes easier and less time-consuming repeat tasks.
- System-level simulation makes easier to manage complexity because it allows to show the interactions between system components.
- Easier to try new ideas with What-If analysis.
- System-level modeling and simulation allows to develop the entire system at the start even though key elements are not known - these elements can be added later without interrupting the development flow.

2.1.3 MODEL-BASED DESIGN APPLICATION STEPS

The main steps of MBD are the following:

1. **Plant modelling** can be data-driven or based on first principles.
   a. *Data-driven plant modelling*, after data collection stage, uses techniques such as System identification: the plant model is identified by acquiring and processing raw data from a real-world system and choosing a mathematical algorithm from which is possible to identify a mathematical model on which is possible to do simulations. This approach is limited by data themselves because they are an expensive resource to get and manage.
   b. *First-principles based modelling* is based on creating a block diagram model that implements known differential-algebraic equations governing plant dynamics, got for instance from literature. A type of first principles-based modelling is physical modelling, where a model consists in connected blocks that represent the physical elements of the actual plant. The limit of this approach is the complexity of these equations.
2. **Controller analysis and synthesis.** The mathematical model conceived in Step 1 is used to identify dynamic characteristics of the plant model. A controller can then be synthesized based on these characteristics.
3. **Offline simulation and real-time simulation.**
4. **Deployment.** Ideally this is done via code generation from the controller developed in Step 2. It is unlikely that the controller will work on the actual system as well as it did in simulation, so an iterative debugging process is carried out by analysing results on the actual target and updating the controller model. Model-based design tools allow all these iterative steps to be performed in a unified visual environment.

2.1.4 TEST TYPES IN MODEL-BASED DESIGN

The step that follows requirements gathering process and simulation process is the testing phase. Indeed, before the developed model is deployed into a specific hardware for production, few verification steps take place (MathWorks, 2019). All the test types are summed up in *Figure 2-5* (Ceppi, 2020). It is important to notice that the arrow points out the crescent level of detail covered by the test.

	Plant	Controller
MIL	Simulation	Simulation
SIL	Simulation	Code (host)
PIL	Simulation	Code (target)
HIL	System part/ real-time sim	Code (target, RT)
RCP	System part/ real-time sim	Real-time sim

Figure 2-5: Model-Based Design Testing

2.1.4.1 MODEL-IN-THE-LOOP

Model-In-the-Loop (*MIL*) is the first step of the testing part in MBD. At this stage, both both the Plant (system to be controlled) and the Controller (algorithm controlling the Plant) are modelled (*Figure 2-6*).

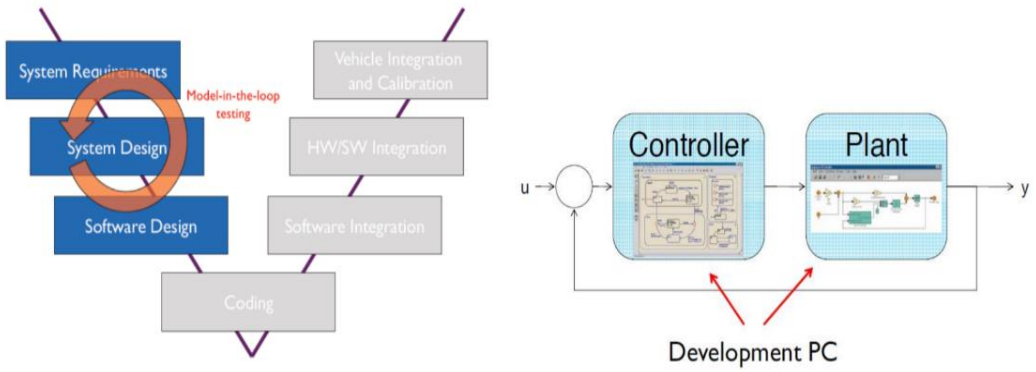

Figure 2-6: MIL testing

2.1.4.2 SOFTWARE-IN-THE-LOOP

Software-In-the-Loop (*SIL*) is the step where the Controller model is replaced by the generated code while the Plant is still simulated. At this stage it is possible to have an idea of whether the control logic - i.e., the Controller model - can be converted to code and if it is hardware implementable. Furthermore, at this stage, is possible to perform a first attempt in Code optimization (*Figure 2-7*).

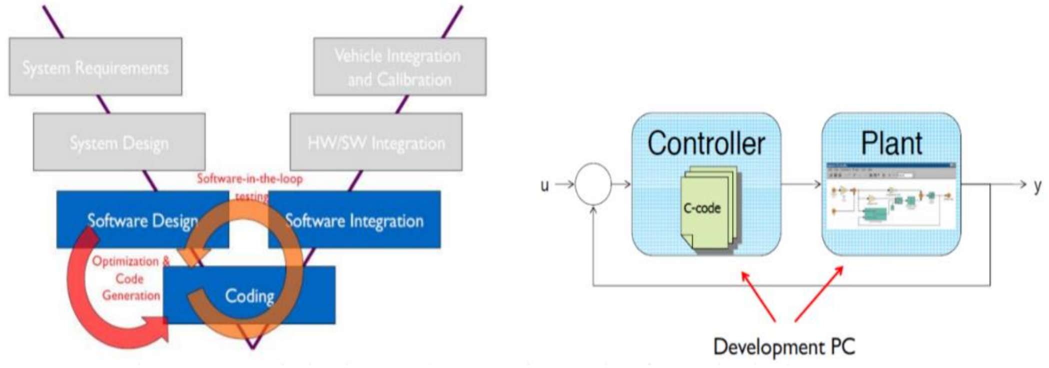

Figure 2-7: SIL testing

2.1.4.3 PROCESSOR-IN-THE-LOOP

Processor-In-the-Loop (*PIL*) is the step where the Controller model generated code is deployed onto an embedded processor/FPGA board and runs a closed-loop simulation with the (still) simulated Plant. This is the first step where the code becomes architecture-specific (

Figure 2-8).

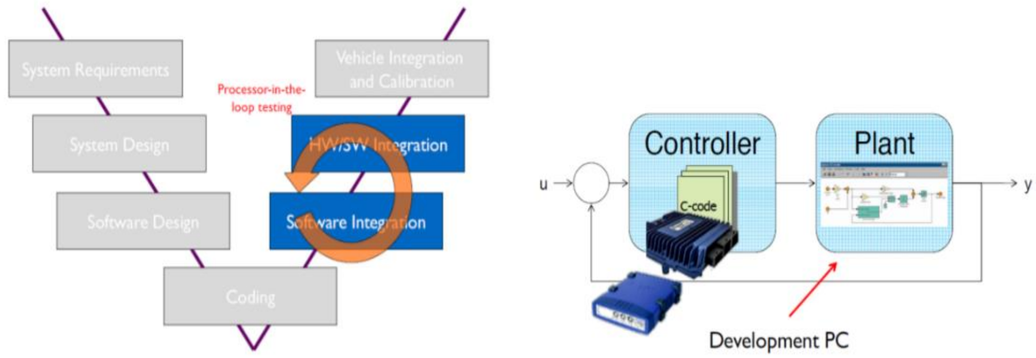

Figure 2-8: PIL testing

2.1.4.4 HARDWARE-IN-THE-LOOP

Hardware-In-the-Loop (*HIL*) is the step where the Controller implementation is co-simulated with the Plant model to test its correctness. The Controller runs on the target hardware, the Plant model on rapid prototyping hardware, and the system is running in real-time. A portion of the model runs in a real-time simulator, and another portion could be implemented as physical hardware (ECU). This phase is suitable to test the interactions with hardware and real-time performance (*Figure 2-9*).

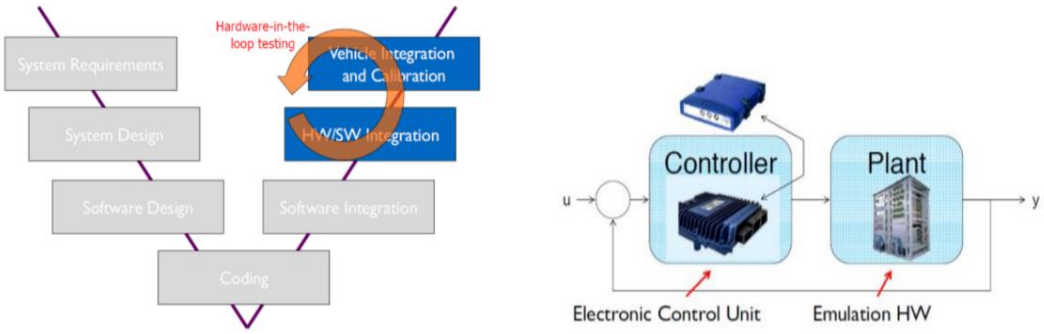

Figure 2-9: HIL testing

2.2 MACHINE LEARNING

In the past few years terms like Artificial Intelligence (AI), Machine Learning (ML), Autonomous Vehicles (AV) and Advanced Driver-Assistance Systems (ADAS) have been dominating tech-talks and headline becoming buzzwords; indeed, in the “age of artificial intelligence” it is not a figment of the imagination to think that AI is being used to complement and enhance car functionalities and behaviours (Eleza, 2019).

This raises a new broader definition of mobility. Driven by the four *ACES trends* - autonomous driving, connected cars, electrified vehicles, and smart mobility - automotive OEMs, suppliers, and new entrants such as tech players and venture capitalists are attempting to build strongholds in the emerging mobility ecosystem. To give a measure of this trend, according to the market analysis driven by

(McKinsey & Company, 2019), by comparing the investment periods 2010-13 and 2014-18, it comes out that up to 29.9 billion dollars have been invested in AV sensors and ADAS components and up to 13.5 billion dollars have been invested in AV software and mapping.

AI is being used to give cars a "degree of automation". According to (SAE International, 2021) there exist 6 levels of driving automation. With focus on the *Level 1*, in this category is possible to find all those systems with the main goal to assist the driver, who is demanded to have the total control of the vehicle. In this scenario, before AI completely takes over the driver's seat, it's being used as a co-pilot to gain the confidence of the users, regulators, and manufacturers. By analyzing data feeds across its sensors, AI can be handy in situations where flesh and blood drivers are prone to making human errors (Eleza, 2019).

Up to now, this work is still missing a definition for *Artificial Intelligence* (*AI*). According to the pioneer of the field, John McCarthy (McCarthy, 2007):

AI is the science and engineering of making intelligent machines, especially intelligent computer programs. It is related to the similar task of using computers to understand human intelligence.

One of the biggest fields of AI is *Machine Learning* (*ML*). According to (Mitchell, 1997):

The field of Machine Learning is concerned with the question of how to construct computer programs that automatically improve with experience.

The general approach is to use some specialized algorithms that build a mathematical model based on sample data, also referred as training data, to make predictions and decisions without being explicitly programmed to do so. Everything starts from the fact that the main goal of a learner is to generalize from its experience: in this context, generalization is the ability of a learning machine to perform accurately on new, unseen examples/tasks after having experienced a learning data set. The training examples come from some generally unknown probability distribution (considered representative of the space of occurrences) and the learner must build a general model about this space that enables it to produce sufficiently accurate predictions in new cases.

In the next section, an overview of the various types of ML algorithms will be introduced.

2.2.1 TYPES OF MACHINE LEARNING ALGORITHMS

There exist several types of ML algorithms. In general, their distinction is done according to (Bishop, 2006):

- Learning approach,
- Type of data they input and output,
- The type of task or problem that they are intended to solve

Figure 2-10 gives an overview of ML algorithms cathegories (Garbade, 2018).

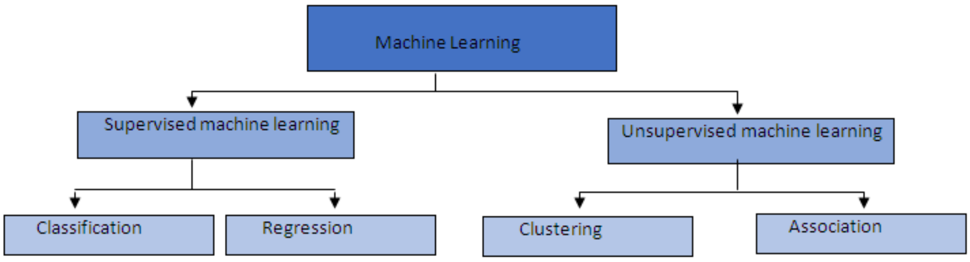

Figure 2-10: Mindmap for ML algorithms

2.2.1.1 SUPERVISED MACHINE LEARNING

Supervised Machine Learning (*SL*) is the task of learning a function that maps an input to an output based on example input-output pairs (Stuart J. Russell, 2010). This is made possible by inferring a function from labeled training data consisting of a set of training examples. Each example consists of a pair of signals: an input vector and a desired output value (also called the supervisory signal or label) (Mehryar Mohri, 2012).

Under the SL umbrella is possible to find *Classification* and *Regression* problems. Both share the same concept of utilizing known datasets (referred to as training datasets) to make predictions. But their similarities end here (Garbade, 2018).

- The main difference between them is that the output variable in regression is numerical (or continuous) while that for classification is categorical (or discrete).
- *Classification algorithms* attempt to estimate the mapping function (f) from the input variables (x) to discrete or categorical output variables (y). An example of classification task might be if a mail message is spam or not.
- *Regression algorithms* attempt to estimate the mapping function (f) from the input variables (x) to numerical or continuous output variables (y). A common example of regression is the prediction of house prices.

Some examples of common Classification algorithms are (Scikit-learn, s.d.):

- C-Support Vector Machines (SVC),
- K-Nearest Neighbours Classification,
- Naïve Bayes,
- Decision trees,
- Multi-layer perceptron (MLP) classifier.

Some examples of Regression algorothms are (Scikit-learn, s.d.):

- R-Support Vector Machines (SVM),
- Bayesian Regression,
- Linear Regression,
- K-Nearest Neighbours Regression,
- Multi-layer perceptron (MLP) regressor.

2.2.1.2 UNSUPERVISED MACHINE LEARNING

Unsupervised learning (*UL*) is a kind of machine learning problem where a model must look for patterns in a dataset with no labels and with minimal human supervision. In UL, only the inputs are available, and a model must look for interesting patterns in the data (Wood, s.d.).

Unsupervised learning problems can be further grouped into clustering and association problems (Brownlee, 2016):

- A *Clustering problem* tries to discover inherent groupings (clusters) in the data, such as grouping customers by purchasing behavior.
- An *Association rule* learning problem tries to discover rules that describe large portions of your data, such as people that buy X also tend to buy Y.

Common examples of UL algorithms are

- K-means clustering,
- Ward's linkage method,
- Gaussian Mixture Models,
- A-priori association algorithms,
- Principal component analysis (PCA).

2.3 PROJECT GOALS AND SPECIFICATIONS

This thesis work aims to study, design, implement and validate a Machine Learning – based Driver Drowsiness Detection (DDD) prototype system using data obtained from ADAS (Advanced Driver Assistant Systems) in simulated conditions by means of a Driver-In-The-Loop (DIL) simulator.

- The study of the system is based on two steps: (1) feasibility study and (2) driver indicators extraction:
    1. The starting point is the Driver Drowsiness Detection problem stated in *Chapter 1*. In this phase of the project, detection methods (described in *Section 1.2 - Drowsiness Detection Methods*) have been studied and analyzed according to their degree of intrusion as well as their feasibility and availability in the simulated environment.
    2. The second step is concerned on the extraction of indicators to evaluate driver physical and driving performance skills.
- The design of the system starts from the core concept of combining several different detection methods with the aim to reach robustness to input signal loss or system failure – the data fusion concept. In this way, if one of the methods fails for any reason, the whole system continues to work properly.

- The implementation step can be divided in two steps as well: (1) algorithm implementation and (2) dataset generation.
    1. The implementation of the algorithm aims to operate a fusion of the various indicators extracted in the previous steps by means of a Machine Learning based algorithm and a Windowed buffer.
    2. The generation of the dataset comprises gathering data through driving sessions as well as some intervention to the simulator during every-day working activities.
- The validation step is done on a subset of the driving sessions (MIL testing).

3 FEASIBILITY STUDY

« It does not matter where you go and what you study, what matters the most is what you share with yourself and the world. »

Santosh Kalwar

In this chapter will be exposed all the steps performed to understand the feasibility of the system and the study related to the extraction of driver indicators used in its implementation. The first part of the chapter will not go into the deep details of the DIL simulator used in Maserati facilities due to the confidentiality of information related to it. Instead, more space will be left to the description of driver sleepiness indicators extraction.

3.1 DRIVER-IN-THE-LOOP SIMULATOR

The source of the whole set of driving data is the Driver-In-The-Loop (DIL) located in *Maserati* facilities. It is a fixed-base driving simulator, constructed starting from the cockpit of a real vehicle.

Simulator features can be listed in this way:

- Active force feedback steering system,
- Automatic transmission (only Neutral can be selected),
- Several ECUs belonging to ADAS domain are equipped on it for everyday company testing activities (e.g., Adaptive Cruise Control system is present),
- All communications and signal transmissions are carried out in real-time using CAN bus protocol to connect ECUs and HIL hardware,

- The visualization system of the simulator uses three video projectors which work together to show the driving scenario on a curved projection screen (180-degrees of vision) outside the vehicle cockpit,
- Inside the cockpit, a set of three cameras paired with infrared sensors is present for driver monitoring operations.

This means that the simulator can provide records for the following input signals:

- Driving variables:
  - Steering wheel angle (counterclockwise positive),
  - Vehicle lateral position over the lane, intended as both the distances between the center and the side of the road,
  - Vehicle speed,
  - Brake pedal position and applied torque,
  - Throttle pedal position and applied torque,
  - Vehicle longitudinal and lateral acceleration,
  - Various cockpit controls actioned by driver (e.g., direction change indicators, ACC engage/disengage).
- Driver variables:
  - Eye gaze,
  - Eye lid,
  - Blink frequency,
  - PERCLOS.

Thus, this simulator assures great flexibility to implement recording systems specifically designed to test the reactions and behaviour of drivers while they are driving with the advantage of providing data from real ECUs.

3.2 SLEEPINESS INDICATORS SELECTION

After the feasibility analysis done by looking at what the simulation environment can provide, next step is to define what signals to use to extract proper indicators that allow to detect driver drowsiness.

3.2.1 RELATED WORK

There are several proposals in literature for what concerns this part of the study. Usually, the starting point is the usage of driver behaviour-based sleepiness indicators without any optimization – that is, a functional form of an indicator from the literature is used but where the parameters that define the indicator are subject to optimization - or generalization process - where a generalized functional form is used for the indicator. Relevant examples in literature are given by the works of (David Sandberg, 2011), (Darrell S. Bowman, 2012), (Eskandarian A. a., 2007) and (Liu, 2009).

For the scope of this work, a set of indicators have been extracted taking as reference the work of (Sandberg, 2011). In the following formulas, x_i represents the i-th sample in a time series, and n denotes the number of samples used when computing the output value of the indicator.

3.2.2 DRIVING INDICATORS

The principle that drives the extraction of indicators is the generalization. One of the problems that needs to be faced in this application is the excessive dependence on the single driver of driving signals. For this reason, sleepiness indicators have been selected according to their degree of generalization from the driving style.

3.2.2.1 STANDARD DEVIATION

The *standard deviation*, denoted by σ, of a time series is defined as:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})}{n-1}} \tag{1}$$

where $\bar{x}$ denotes the arithmetic mean of the time series. It will be applied to extract the following indicators from the driving signals:

- ***Standard deviation of the lateral position.*** It is one of the most popular sleepiness indicators in driver drowsiness detection studies. It is used to describe *lane-drift frequency*.

- ***Standard deviation of the steering-wheel angle.*** Steering wheel inputs can be an indicator of drowsy driver impairment. Alert drivers will respond to lane deviations early with many small amplitude steering movements which correct the car's trajectory, whereas drowsy drivers will respond slowly to lane deviations and make large steering wheel movements to correct for larger lane deviations.

3.2.2.2 TIME-TO-LANE CROSSING (TLC)

Time-to-Lane Crossing (*TLC*) measure is defined as the time needed for the vehicle to cross either lane limit (left or right), given the current vehicle speed and acceleration, road curvature, lateral position, and direction of the vehicle. It is already provided by the simulating environment.

3.2.2.3 RAPID STEERING-WHEEL MOVEMENT

Rapid Steering-Wheel Movement (*RSWM*) measures the fraction that the steering-wheel velocity exceeds a specified threshold value during a given time interval. Using the derivative of the SWA signal, which is denoted here as $\dot{s}$, the indicator is computed as:

$$RSWM = \frac{1}{n-1}\sum_{i=2}^{n} h(\dot{s}_i) \qquad \textbf{(2)}$$

where:

$$h(\dot{s}_i) = \begin{cases} 1, & if\ |\dot{s}_i| > d \\ 0, & otherwise. \end{cases} \qquad \textbf{(3)}$$

Here *d* is a specified threshold set to the value of 13 °/s as proposed in (Garcia Daza, 2014).

3.2.3 DRIVER INDICATORS

3.2.3.1 PERCENTAGE EYE OPENNESS TRACKING

Percentage Eye Openness Tracking (*PERCLOS*) as previously stated, measures duration of blinks and eye closures, and proportion of time eyes closed over a specified time interval. In this work, a period of 60 s has been selected.

4 SYSTEM DESIGN AND IMPLEMENTATION

«In an era of stress and anxiety, when the present seems unstable and the future unlikely, the natural response is to retreat and withdraw from reality, taking recourse either in fantasies of the future or in modified visions of a half-imagined past. »

Alan Moore, Watchmen

In this chapter will be explained decisions made for the design and implementation of the proposed DDD system. The starting point will be the concept of data fusion to enhance system robustness. Then, there will be insight on how to reach in practice data fusion concept.

4.1 DATA FUSION CONCEPT

In *Section 1.2* various methods to detect driver drowsiness have been introduced. Furthermore, in *Section 1.3* various commercial solutions have been illustrated. What comes out is that drive drowsiness detection systems, in their largest part, rely on a single method to detect driver sleepiness. Various factors may cause such a system to easily fail and stop working correctly with consequent unreliable results, as already proved in literature by (Samiee, 2014) and (Darrell S. Bowman, 2012). A non comprehensive list of possible causes of failure is given in *Table 1*:

Method	***Restrictions***
Driver's face tracking **(Image processing methods)**	1. Performance reduction in low ambient light,

	2. Performance reduction while wearing glasses, having beard, wearing a mask, 3. Tracking failure due to fast movements.
Heart and brain signal processing	1. Intrusive method, 2. Reduce driver concentration, 3. Driver may forget to use (wear) the sensors.
Driver reaction to a message	1. Intrusive method 2. Not real-time
Lane departure warning	1. Unnecessary warnings 2. System failure due to lack of clarity in road markings
Driver vehicle interaction/driving behavior	1. Closely dependent to driver's driving habits 2. Changes by driver emotional state (anger, anxiety, sadness)

Table 1: List of the different drowsiness detection methods and their possible cause of failure.

In this work, the objective of the developed detection system is to alleviate some of the issues associated with a single-measure device thus it does not rely on one method hence, if one of the methods fails for any reason, the whole system will continue to work correctly. System accuracy is slightly reduced in such a circumstance, but the detection system will be more robust and, hence, more reliable.

To choose a good combination from different available methods, together with feasibility constraints reported in *Chapter 3* and, to utilize the benefits of the methods of the different categories, an algorithm based on image-processing and driver-vehicle interaction is proposed:

- Image processing method: *PERCLOS* metric,

- Driver-vehicle interaction methods:
    1. Vehicle lane position metrics,
    2. Steering-wheel metrics.

Figure 4-1 shows a diagram of the DDD prototype data fusion workflow for reference.

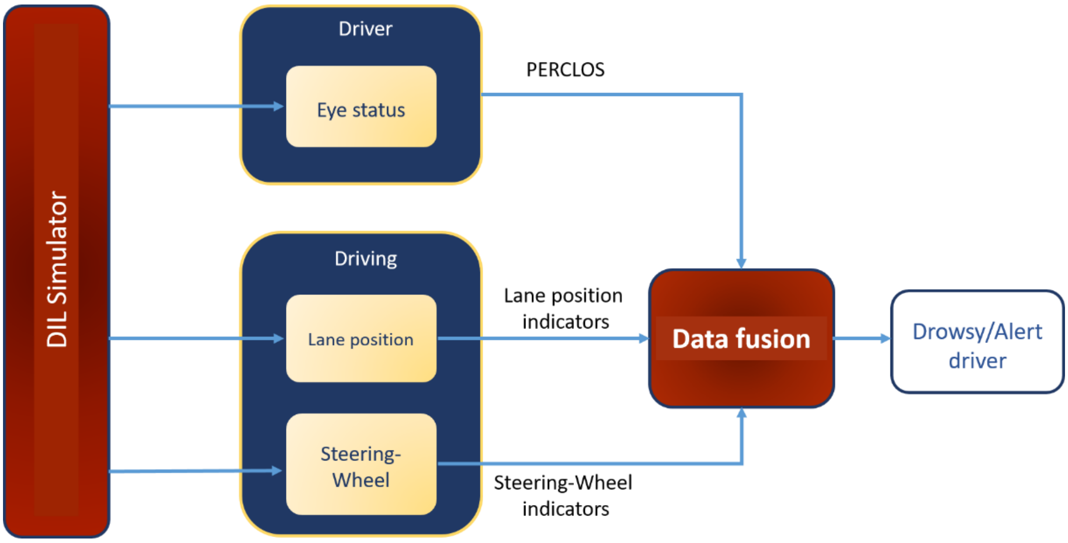

Figure 4-1: Data Fusion workflow

4.2 DATA FUSION METHODOLOGY

There are two important key tasks associated with data fusion practical implementation: (1) data fusion algorithm derivation and (2) data acquisition.

4.2.1 DATA FUSION ALGORITHM

The data fusion and so also the drowsiness detection is based on the learning and prediction given by a Machine Learning algorithm. Gathered information by each of three detection methods is first processed using a specific classifier. Hence, each branch of the system can distinguish the driver's level of drowsiness, independently.

4.2.1.1 MACHINE LEARNING PROBLEM STATEMENT

To quickly understand the elements the following framework has been used (Brownlee, 2013). The framework involves answering three questions to varying degrees of thoroughness:

- *Step 1*: What is the problem?
- *Step 2*: Why does the problem need to be solved?
- *Step 3*: How would I solve the problem?

For the *Step 1* a first informal description can be given as follows:

There is the need to define a Driver Drowsiness Detection system (DDD) able to recognize if the driver is alert or drowsy.

After that, it is important to give a formal definition to the problem defined above (Mitchell, 1997):

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

This formalism can be applied to the object of this work in this way:

- *Task* (*T*): Recognize a drowsy or alert driver (Binary classification problem).
- *Experience* (*E*): A series of drives where some drivers are drowsy and some are not,
- *Performance* (*P*): Classification accuracy, the number of drowsy and alert drivers predicted correctly out of all drives considered as a percentage.

Motivations required in *Step 2* have been already discussed in *Section 1.1*. For what concerns the *Step 3*, it will be exploited in the next sections.

4.2.1.2 SYSTEM ARCHITECTURE

The system architecture can be detailed in the following way:

- Information coming from each of the detection methods is first processed using a specific classifier, hence each branch of the system can distinguish if the driver is drowsy or not independently.
- Downstream from the classifiers, there are three different weighting units which assign a different gain to the classifiers' outputs.
- There is a *Signal Availabilty Unit* (*SAU*) which has a discrete output value of 0 or 1 serving as an indicator for input signal availability. In case of

unavailable signal, the corresponding classifier should not influence the drowsiness indicator extraction.

Figure 4-2 shows the general architecture of the proposed system.

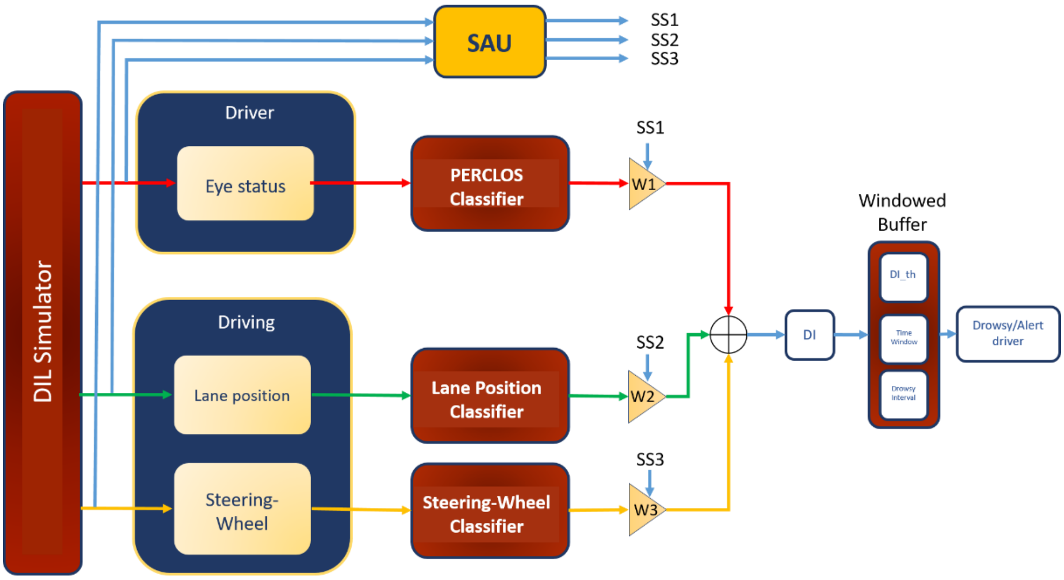

Figure 4-2: Architecture of the proposed DDD system

In this work the data fusion and so the drowsiness detection has been exploited as a binary classification problem, but it is possible to use a multiclass classifier as well as a regressor to reach the same goal.

4.2.2 DROWSINESS INDICATOR AND WINDOWED BUFFER

Independently from the used classification algorithm, the final output of the system is a *Drowsiness Indicator* (*DI*). It is composed by the linear combination of three components:

- The value of each SS-i output because the final value must be dependent on the currently available systems and, as stated in *Section 4.2.1.2*, the corresponding subsystem should not influence the drowsiness indicator value in case it is not working.
- The output of each of the detection subsystems (O_i),
- The weight assigned to each detection subsystem (w_i). This value is extracted as a function of the whole system accuracy and the sub-system accuracy.

Analitically:

$$SS1 * w_1 + SS2 * w_2 + SS3 * w_3 = 1 \quad (4)$$

$$w_i = \frac{i - th\ method\ accuracy}{\sum_{i=1}^{3} i - th\ method\ accuracy}, i = 1, 2, 3 \quad (5)$$

$$DI = SS1 \cdot w_1 \cdot O_1 + SS2 \cdot w_2 \cdot O_2 + SS3 \cdot w_3 \cdot O_3 \quad (6)$$

This indicator, however, might make the system too much susceptible to external factors like noise since it is related to a single sample. For this reason, a *Windowed Buffer* - based on a fixed time window and a threshold value - has been inserted downstream the system. It can be detailed according to the following pseudo-code (*Table 2*).

1. Get ***DI*** indicator value for a single sample;
2. If ***DI*** is *larger or equal* to a certain threshold (***DI_th***):

   Drowsy sample;
3. Else:

   Alert sample;
4. In the last ***Time_window*** seconds → If the driver is detected drowsy for *more than* ***Drowsy_interval*** seconds:

   Drowsy driver;
5. Else:

   Alert driver;

Table 2: Windowed Buffering pseudocode

While *DI_th* and *Time_window* parameters are considered as arbitrary fixed, the *Drowsy_window* – amount of time needed by the system to consider the driver as drowsy - is linked to the Time_window parameter by the Eq. (7):

$$Drowsy\_interval = \text{P\% of Time\_window} \quad (7)$$

4.2.2.1 CLASSIFICATION ALGORITHM

The goal of the classifiers for each of the methods branches is to classify the driver's state into two different possible classes: alert or drowsy. In this work, *K-Nearest Neighbors* (*K-NN*) *classifiers* have been used. The features were normalized and labeled as either alert (0) or drowsy (1) before proceeding to build each of the classifiers.

K-NN algorithm has been choosen for its widely usage in industry classification problems (Srivastava, 2018). The reason for this widespread usage is due to its simplicity and very low calculation time. The algorithm can be broken down into writing 3 primary functions (Latysheva, 2017):

1. Calculate the distance between any two points,
2. Find the nearest neighbours based on these pairwise distances,
3. Majority vote on a class labels based on the nearest neighbour list.

Figure 4-3 gives an insight on the algorithm intermediate steps (Cambridge Coding Academy, 2016).

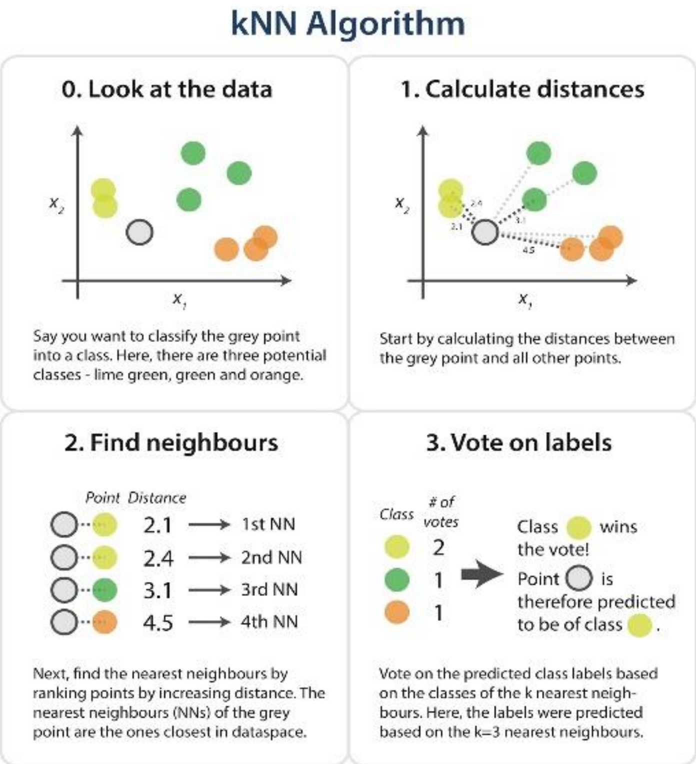

Figure 4-3: K-NN algorithm details

4.2.3 DATA ACQUISITION AND PREPARATION

The dataset used to train classifiers is composed of several sequences collected during several test drives on a DIL simulator (*Section 3.1*). Due to privacy and confidentiality issues, a lot of information about driving sessions partecipants will be omitted.

Drives were conducted in both daytime and nighttime. Details regarding testing conditions and driving path can not be disclosed. Data sampling time for both CAN bus and driver-related signals is 10 ms.

4.2.3.1 GROUND TRUTH GENERATION

During driving sessions, for each driving interval the driver was asked to insert a value ranging between 1 and 9 to categorize data collected in that driving interval. As already discussed in *Section 1.2.2*, the *Karolinska Sleepiness Scale* (*KSS*) is the most used subjective scale for drowsiness evaluation and has been used to categorize data during driving sessions. There are several reasons to discard this kind of reporting, also according to literature (Schmidt, et al., 2009):

1. It is considered as an intrusive method due to the interaction with external entities of the driver,
2. Driver must be trained to correctly recognize his current state,
3. After several hours of driving, the driver might have difficulty in judging their physical state, introducing wrong evaluations.

There are also other authors, like (Akerstedt, 1990) and (Sandberg, 2011), that totally discard those arguments and suggest the usage of KSS as subjective report method.

In this work, to avoid as much as possible driver interactions with external distracting elements, an automated approach is used. During each of the drives, the DIL simulator has been equipped with a *KSS logging application* properly designed and implemented. Its workflow is illustrated in *Figure 4-4*.

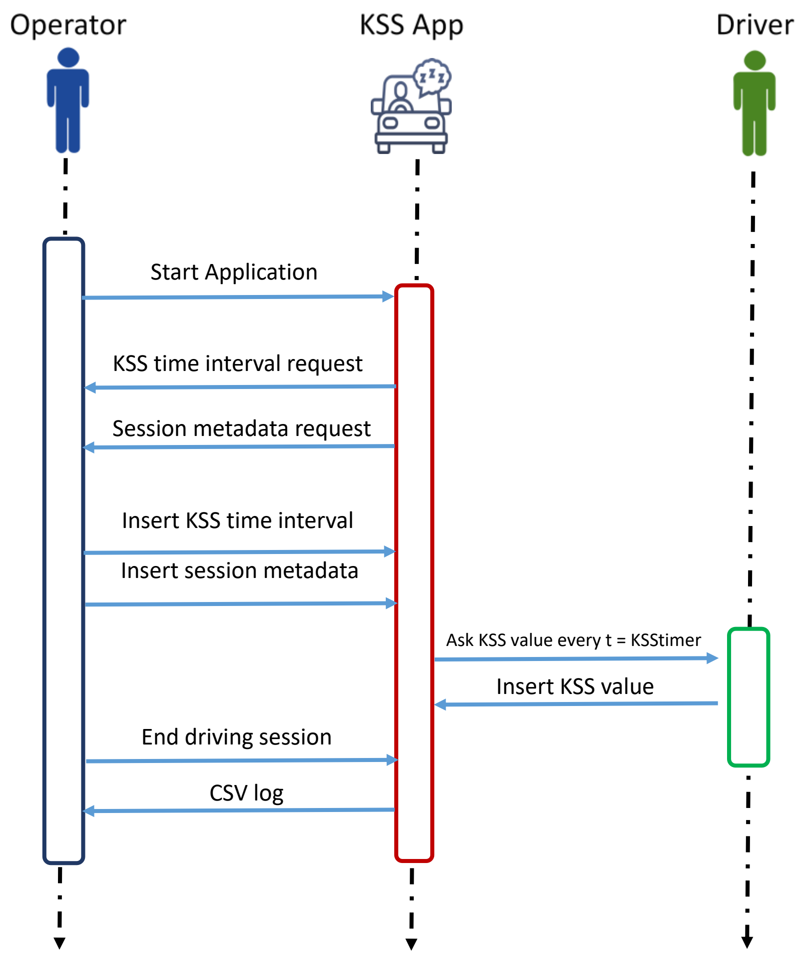

Figure 4-4: KSS Log App sequence diagram

It is important to notice how the usage of this application minimizes direct interactions between the operator and the driver by acting as a sort of "filter" between them.

4.2.3.2 DATA PRE-PROCESSING

Machine Learning algorithms learn from data so the step of preparing data to feed them is critical because the quality of data and the useful information that can be derived from it directly affects the ability of our model to learn (Kumar, 2018).

According to the system structure illustrated in *Section 4.2.1.2*, because there are 3 classifiers to train this means that there are also 3 different datasets to deal with thus in the next subsections the various techniques, adapted to each dataset, will be introduced.

4.2.3.2.1 BASIC CLEANING

The first step is to *remove null values* that might be present among the samples present in the dataset. They must be removed because a model able to handle this kind of values doesn't exist.

Another common procedure to apply on datasets is to remove *invalid values*: an example of invalid value that has been encountered is the KSS value equal to -9999. That value is related to those samples which dataset value might be not present or not valid.

These important cleaning procedures have been applied on all the 3 datasets.

4.2.3.2.2 FEATURE ENGINEERING

Feature engineering is the process of selecting, manipulating, and transforming raw data into features that can be used in supervised learning (Patel, 2021). A feature is any measurable input that can be used in a predictive model and in this work features that will be used are those listed in *Section 3.2.2*. However, in that section the only feature selection step has been performed. What is missing is the manipulation and transformation steps. After some *Exploratory Data Analysis* (*EDA*), those are the performed actions on datasets:

- ***Windowing.*** Indicators have been calculated by applying windowing techniques over the input signals covering an interval of 20 s to generate some time series.
- ***Data scaling.*** To have data that are on the same scale and avoid wrong weighting from the model in the training phase, standardization and normalization scaling techniques are used:
    - ***Data standardization.*** It is the process of scaling values while accounting for standard deviation. With this step, dataset values are transformed such that their mean is null and standard deviation is 1.

$$z = \frac{x_i - \bar{x}}{\sigma} \quad \textbf{(8)}$$

Standardization has been applied, after testing, to Steering-Wheel dataset and PERCLOS dataset due to better performances.

- ***Data normalization.*** It is the process of scaling all values in a specified range.

$$x_{norm} = \frac{x_i - x_{min}}{x_{max} - x_{min}} \quad \textbf{(9)}$$

Normalization has been applied to Vehicle Lane Position dataset.

- ***Data transformation.*** To train models on data that are Gaussian shaped, data transformation has been performed.
  - ***Log-transformation.*** This kind of transformation helps reducing data skewness. This transformation has been applied on PERCLOS dataset (Ma, 2019).
  - ***Box-Cox power transform.*** This technique is used to stabilize variance and make Vehicle Lane Position data more normal distributed-like. It is defined according to:

$$y_i^{(\lambda)} = \begin{cases} \dfrac{y_i^{\lambda} - 1}{\lambda} & \text{if } \lambda \neq 0, \\ \ln(y_i) & \text{if } \lambda = 0, \end{cases} \quad \textbf{(10)}$$

- ***Removing Road curvature effect.*** According to several works in literature, like (Eskandarian A. S., 2007) or (Li, 2017) the steering angle signal contains two different types of waveforms: one for straight lines and one for curves. In the straight sections, the waveforms consist only of the constant steering adjustments to keep the vehicle in the center of the lane, while in the curving sections, the waveforms contain road curvature, in addition to the component of steering adjustments for lane keeping. To remove this effect, a simple subtraction of the trend from the signal has been performed.

5 SYSTEM VALIDATION AND RESULTS

« I can't change the direction of the wind, but I can adjust my sails to always reach my destination. »

Jimmy Dean

In this chapter will be detailed the training and validation procedures performed on the system. First, for each branch of the system will be reported the training results, together with the tuning parameters chosen. Finally, the final system validation results will be reported and reviewed.

5.1 SINGLE METHODS TRAINING RESULTS

The K-NN classifier for each of the 3 methods was trained by using driver and driving sleepiness indicators detailed in *Section 3.2* after the data processing procedures explained in *Section 4.2.3*. Next sections will be used to expose training results in terms of overall accuracy together with the tuning parameters chosen.

The evaluation is done according to the following parameters, all expressed as a percentage:

- *True Positives (TP).* Performance of the system in correctly identifying a drowsy driver.
- *False Positives (FP).* System failure rate in identifying an alert driver as a drowsy one.
- *False Negatives (FN).* System failure rate in identifying a drowsy driver as an alert one.
- *True Negatives (TN).* System success rate in identifying a drowsy driver.

- *Classification Accuracy*. Ratio of number of correct predictions to the total number of input samples.

Futhermore, the number of samples categorized as drowsy and awake will be reported as well.

5.1.1 PERCLOS CLASSIFIER PERFORMANCES

Driver Status	Alert	Drowsy	TN	FN	FP	TP	Accuracy
Alert	877499	417768	67.74%	32.26%	60.8%	39.2%	54.69%
Drowsy	660905	424763					

Table 3: PERCLOS Classifier training results

Training Hyperparameters:

- *Neighbours*: 5,
- *Weights*: uniform,
- *Distance:* Euclidean.

5.1.2 VEHICLE LANE POSITION CLASSIFIER PERFORMANCES

Driver Status	Alert	Drowsy	TN	FN	FP	TP	Accuracy
Alert	1088386	145758	88.1%	11.9%	16.28%	83.72%	86.1%
Drowsy	169372	870893					

Table 4: Vehicle Lane Position Classifier training results

Training Hyperparameters:

- *Neighbours*: 50,
- *Weights*: uniform,
- *Distance:* Euclidean.

5.1.3 STEERING-WHEEL ANGLE CLASSIFIER PERFORMANCES

Driver Status	Alert	Drowsy	TN	FN	FP	TP	Accuracy
Alert	1472942	411716	78.2%	21.8%	29%	71%	74.86%
Drowsy	459258	1120682					

Table 5: Steering-Wheel Angle Classifier training results

Training Hyperparameters:

- *Neighbours*: 5,
- *Weights*: uniform,
- *Distance:* Euclidean.

5.2 FINAL SYSTEM VALIDATION AND RESULTS

Classifiers described in the *Section 5.1* are merged together in a system with the structure detailed in *Section 4.2.1.2* in order to properly fuse data. The output of the system is a *Drowsiness Indicator* (*DI*), extracted according to the pseudocode procedure described in *Table 2* and the analytical formulas presented in *Section 4.2.2*.

The evaluation of the system is done according to the following parameters:

- *Signal status (SS-i).* This parameter is used to check the subsystems are working (1) or not (0) to exclude them or not from the final output. To recall:
    - *SS1* is the check referred to the PERCLOS subsystem.
    - *SS2* is the check referred to the Lane Position subsystem.
    - *SS3* is the the check referred to the Steering-Wheel Angle subsystem.
- *Weights.* The weight assigned to each detection subsystem (w_i).
    - w_1 is the weight associated to the PERCLOS subsystem.
    - w_2 is the weight associated to the Lane Position subsystem.
    - w_3 is the weight associated to the Steering-Wheel Angle subsystem.

- *Drowsiness Indicator Threshold* (*DI_Th*). This threshold is used to evaluate wheter the driver is alert or drowsy starting from the *DI*.
- *Time Window*. Time interval used to bufferize system output.
- *Drowsy Interval*. Time needed for the driver to be considered as drowsy.
- *Accuracy*. Ratio of number of correct predictions to the total number of input time windows.

5.2.1 DROWSY DRIVE RESULTS

Scenario	Signal Status			Weights			Time Window	Drowsy Interval
	SS1	*SS2*	*SS3*	w_1	w_2	w_3		
1	1	1	1	0.253	0.399	0.347	20 s	0.3
2	1	1	0	0.388	0.611	0	20 s	0.3
3	1	0	1	0.422	0	0.577	20 s	0.3
4	0	1	1	0	0.535	0.4650	20 s	0.3

Table 6: Scenarios for DI_th evaluation

5.2.1.1 DI_TH EVALUATION

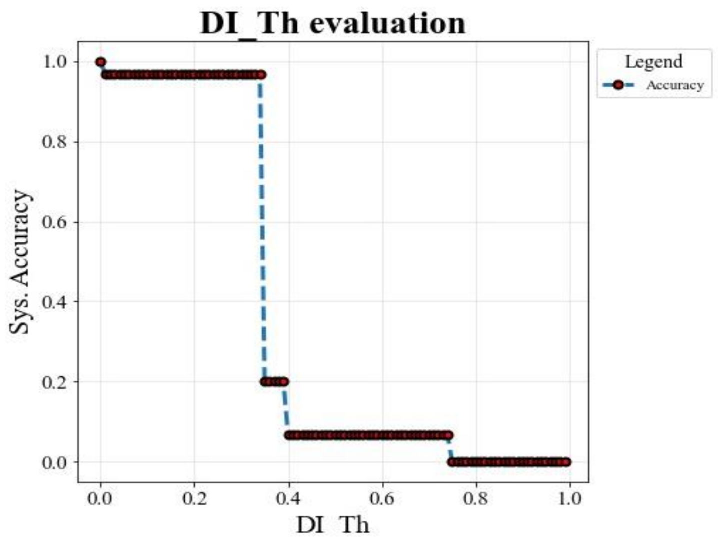

Figure 5-1: DI_Th evaluation in Scenario 1

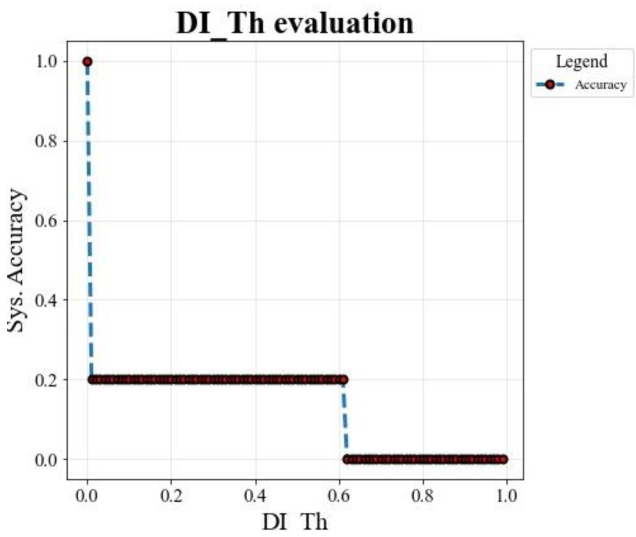

Figure 5-4: DI_Th evaluation in Scenario 2

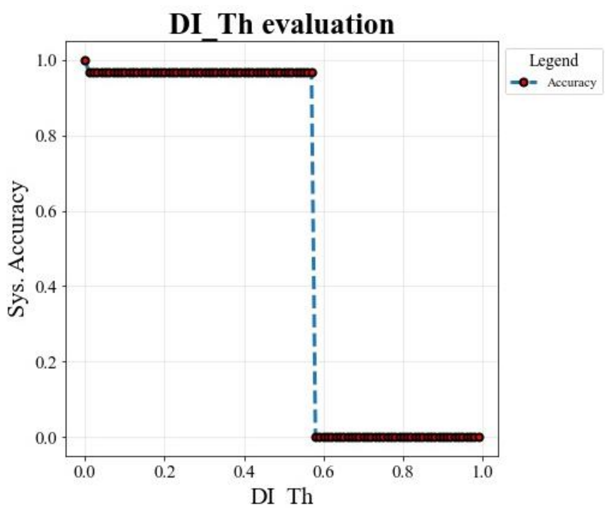

Figure 5-3: DI_Th evaluation in Scenario 3

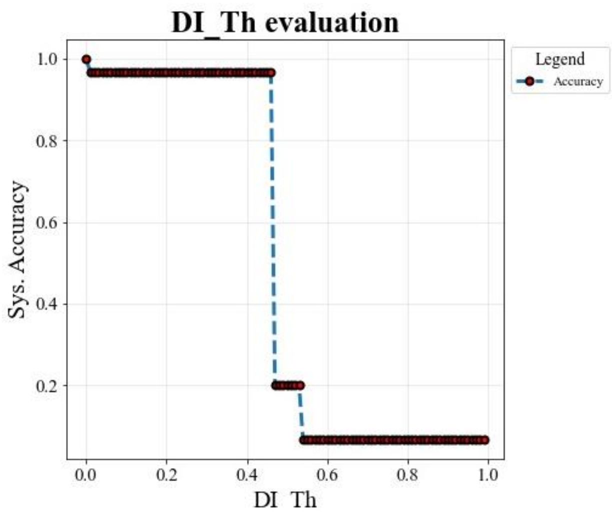

Figure 5-2: DI_Th evaluation in Scenario 4

5.2.1.2 TIME WINDOW EVALUATION

Scenario	Signal Status			Weights			DI_Th	Drowsy Interval
	SS1	*SS2*	*SS3*	w_1	w_2	w_3		
1	1	1	1	0.253	0.399	0.347	0.3	0.3
2	1	1	0	0.388	0.611	0	0.3	0.3
3	1	0	1	0.422	0	0.577	0.3	0.3
4	0	1	1	0	0.535	0.4650	0.3	0.3

Table 7: Scenarios for Time window evaluation

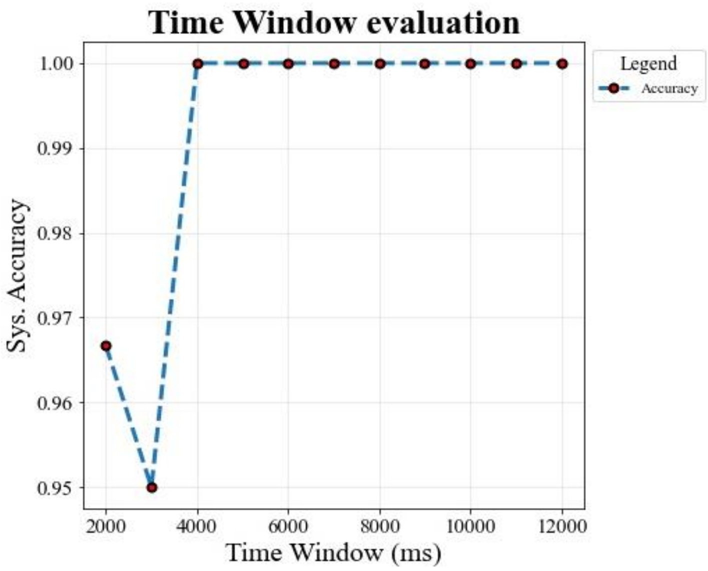

Figure 5-5: Time Window evaluation in Scenario 1

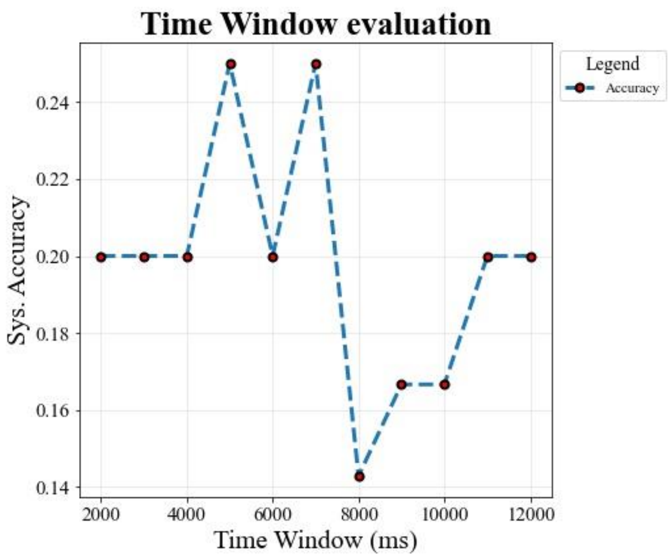

Figure 5-6: Time Window evaluation in Scenario 2

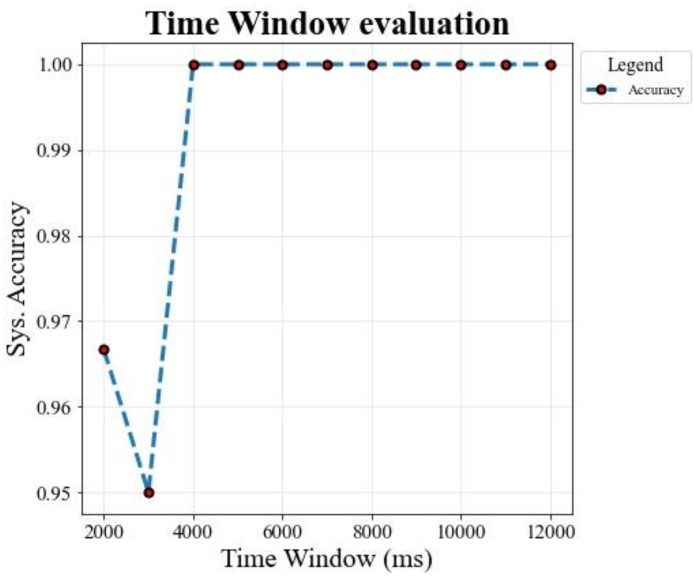

Figure 5-8: Time Window evaluation in Scenario 3

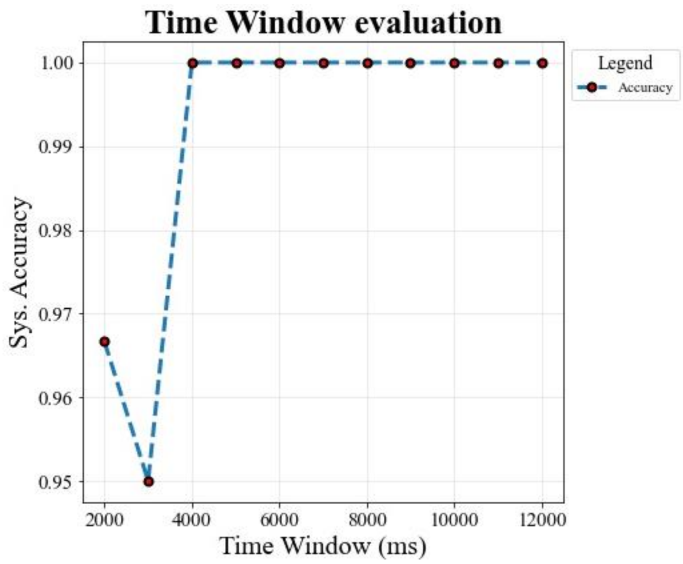

Figure 5-7: Time Window evaluation in Scenario 4

5.2.1.3 DROWSY INTERVAL EVALUATION

Scenario	Signal Status			Weights			DI_Th	Time Window
	SS1	***SS2***	***SS3***	w_1	w_2	w_3		
1	1	1	1	0.253	0.399	0.347	0.3	20 s
2	1	1	0	0.388	0.611	0	0.3	20 s
3	1	0	1	0.422	0	0.577	0.3	20 s
4	0	1	1	0	0.535	0.4650	0.3	20 s

Table 8: Scenarios for Drowsy interval evaluation

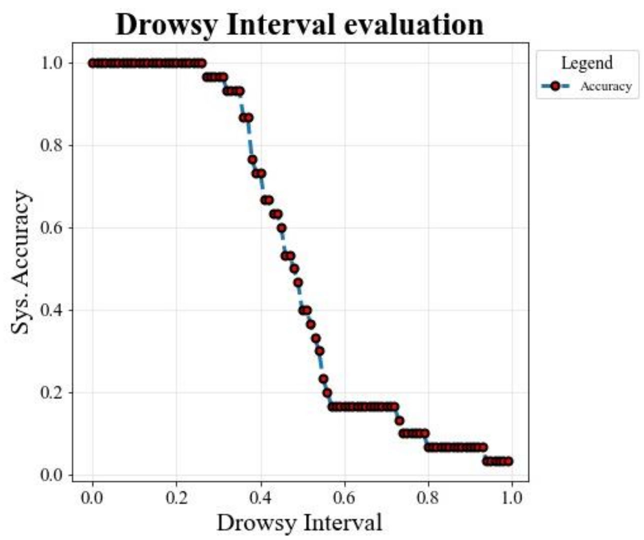

Figure 5-9: Drowsy Interval evaluation in Scenario 1

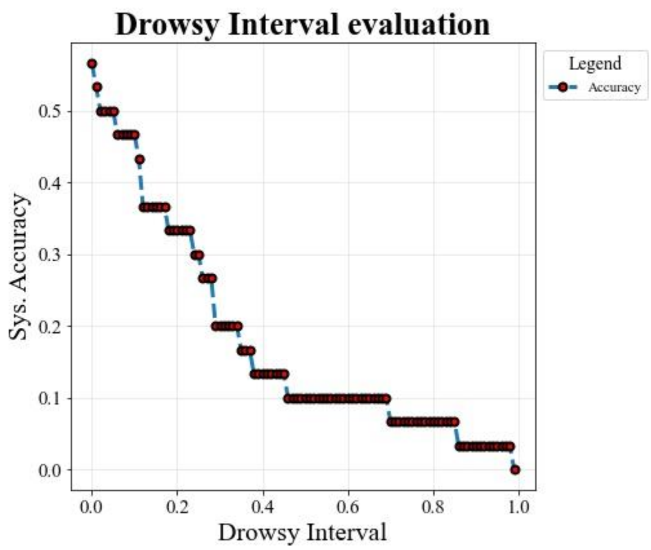

Figure 5-10: Drowsy Interval evaluation in Scenario 2

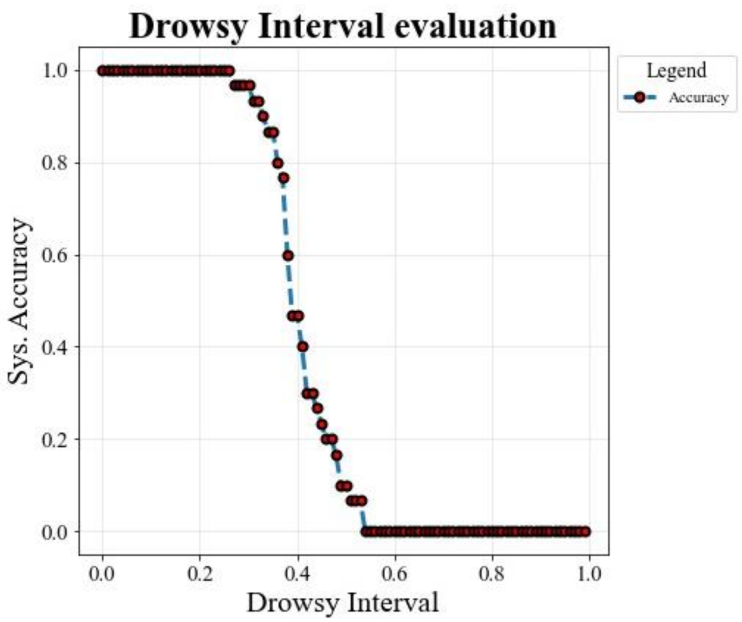

Figure 5-12: Drowsy Interval evaluation in Scenario 3

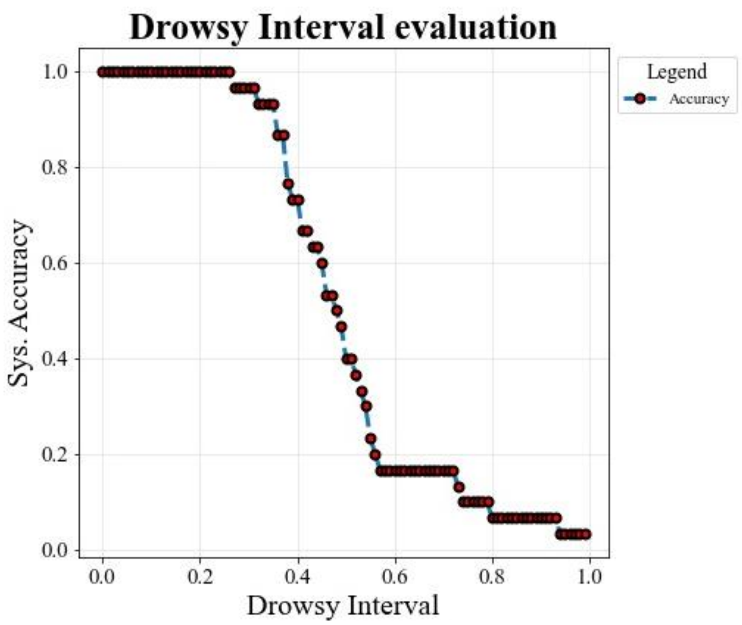

Figure 5-11: Drowsy Interval evaluation in Scenario 4

5.2.2 ABOUT THE RESULTS

From the previous sections comes out that overall the system can detect, by properly tuning its parameters, correctly a drowsy driver, even overcoming signal noise and losses. It is possible to conclude that the system grants better results compared to the performance of each sole method.

However, previous results tell something also about the system trade-offs that must be considered:

- With reference to the *DI_Th* parameter section, the trend that comes out from the graphs referred to each of the scenarios is the high system accuracy for low values of the parameter. This introduces an important trade-off to consider: the lower *DI_Th*, the higher the system accuracy and the higher its intrusivity as well. A particular mention must be made about the second scenario: it is important to notice the abrupt accuracy degradation for values higher than zero. This problem can be solved by tuning the other parameters properly.
- For what concerns the *Time Window*, the good news for almost all the scenarios is the higher system accuracy for higher values of the window time.
- Finally, talking about the *Drowsy Interval*, looking at all the graphs for each of the test scenarios it is possible to notice the same trend of the *DI_th* section: the lower the value and the higher is the system accuracy. The price to pay is also the higher intrusivity of the system. The difference between the results related to these parameters is the higher flexibility in terms of tuning parameters choice granted by the Drowsy Interval parameter.

6 FUTURE STEPS AND FINAL THOUGHTS

« Take hold of the future or the future will take hold of you - be futurewise. »

Patrick Dixon

This work proposed a Driver Drowsiness Detection system based on the fusion of data gathered from three different drowsiness detection methods – PERCLOS, Vehicle Lane Position and Steering-Wheel angle. The system can grant robustness against signal noise and losses, even in the case where one of the methods fails or is providing unreliable data. Data coming from each of the three detection methods have been fused with the joint action of a Machine Learning algorithm – K-Nearest Neighbours – and a Windowed Buffering unit, tunable with three different parameters – Drowsiness Indicator Threshold, Time Window and Drowsiness Interval.

According to the obtained results, the following concluding remarks can be drawn:

- *Accuracy* – the system, overall, can detect a drowsy driver with a very high accuracy, depending on the choice of the tuning parameters. The trade-off that must be respected is the one about accuracy/sensitivity of the system. Sensitivity is increased to the extent that error variation (e.g., false positives/negatives) is reduced. The goal is to properly tune the system to not have a high number of errors – so that they do not reduce the driver's trust of the system - and an acceptable level of sensitivity without being too much intrusive.
- *Adaptability to Various Environmental Conditions* – thanks to the data fusion the system can adapt itself to various environmental conditions, both internal to the cab - it is not possible to properly detect driver's eyes by

means of cockpit cameras due to ambient lighting – and external to the cab – high noise levels, road geometry, resilience in case of system fault. This grants to the system, in principle, a high level of *generalizability*.

- *Adaptability to Various Driver Physical Characteristics* – The system can accommodate a wide range of physical characteristics of the driver, including demographic features – e.g., age and gender – and physical features.

6.1 FUTURE WORK

Drowsiness detection is a hard task due to the difficulty to properly detect drowsiness effects on humans or to properly generalize their effects to other humans. These problems have been considered during the genesis of this work in several ways – e.g., driver and driving indicators extraction, data analysis and feature engineering. Although the results are positive, several improvements can be done:

- *Gathering more and better data* – looking at the classifiers, their performances alone are too low, especially the PERCLOS one. Among the reasons of these poor performances there is for sure something wrong in the data gathering and analysis process so the aim for future works is to have a more accurate system by advising and solving errors and wrong assumptions made in this phase.
- *Examine other non-intrusive methods and new drowsiness indicators* – methods used in this work are not the only non-intrusive ones. Same goes for drowsiness indicators. There are other possibilities that can grant a better combination of drowsiness detection methods, both on the driver and driving side, so the aim for future work is a deeper examination of these possible combinations.
- *Examine new Machine Learning algorithms* – in this work classifiers are based on the K-NN algorithm because it granted on the gathered data the best performances on the classification task paired with the shortest training time. System structure grants flexibility on the choice of the classifiers algorithm so, although a neural network-based solution has already been

evaluated and discarded due to poor classification performances, the aim is to further explore this ML field.

- *Make the system available in real-time and outside a simulated environment* – as it is the system is basically at its start with the goal to prove in principle its effectiveness in fighting signals losses, noise, and system failures. However, there is still too much work to be done in terms of optimization, tuning and performance improvements with the goal of having a system able to operate in real-time and in a real-world scenario.

ACKNOWLEDGEMENTS

« I would maintain that thanks are the highest form of thought, and that gratitude is happiness doubled by wonder. »

G.K. Chesterton

At the end of this work and more in general of this phase of my life and career I want to dedicate this space to those people who were fundamental in helping and supporting me during this journey.

I would like to express my sincere gratitude to my supervisors *Prof. Carlo Concari*, *Dr. Roberto Rabbeni* and *Ing. Gianfranco Bordonaro* for their invaluable supervision, support and tutelage during my academic career and my internship.

Special thanks go to the company *Maserati S.p.a.* for the internship opportunity and to the *Virtual Engineering* team, particularly to Davide, Matteo, Chiara and Francesco for their help, support and the good time spent together.

A special thanks goes to my beloved partner Elena, patient, severe, playful guide in these hard years but never too much thanks to her presence, her comfort, her irreducible tenacity in encouraging me to "never give up" and for always showing me the world like a better place despite everything.

I don't know if I find the right words to thank and express my gratitude to, last but not less important, my parents and my family. I would like this achievement of mine, as far as possible, to be a reward for them too and for the sacrifices they have made. My gratitude to you will never be enough for being there, for patiently supporting me, for teaching me what is "right" and what is not, for never let me lose the way. Without you, today's Ivan Enzo Gargano certainly wouldn't be who he is. Thank you for everithing.

Thanks for your love.

Bologna, 21st March 2022

Ivan Enzo Gargano

REFERENCES

Aarenstrup, R. (2015). *Managing Model-Based Design.* The MathWorks, Inc. Retrieved February 11th, 2022, from https://it.mathworks.com/campaigns/offers/managing-model-based-design-auto.html

Abdulbari Bener, E. Y. (2017). Driver sleepiness, fatigue, careless behavior and risk of motor vehicle crash and injury: Population based case and control study. *Journal of Traffic and Transportation Engineering (English edition)*. doi:https://doi.org/10.1016/j.jtte.2017.07.005

AISIN. (2021, June). *AISIN Develops New Products for Enabling Advanced Driving Assist Technology — Adopted in TOYOTA "MIRAI" and LEXUS "LS"*. Retrieved February 11th, 2022, from www.aisin.com: https://www.aisin.com/en/news/2021/005406.html

Akerstedt, T. &. (1990). Subjective and objective sleepiness in the active individual. *The International journal of neuroscience*. doi:https://doi.org/10.3109/00207459008994241

Al-Mekhlafi, A.-B. &. (2020). THE RELATIONSHIP BETWEEN FATIGUE AND DRIVING PERFORMANCE: A REVIEW AND DIRECTIONS FOR FUTURE RESEARCH. *Journal of Critical Reviews*. doi:10.31838/jcr.07.14.24

Automotive Driver Drowsiness Detection System Market Size, Share & Industry Analysis. (2022). Retrieved February 11th, 2022, from Fortune Business Insights: https://www.fortunebusinessinsights.com/automotive-driver-drowsiness-detection-system-market-103620

Baek, H. J. (2009). Nonintrusive biological signal monitoring in a car to evaluate a driver's stress and health state. *Telemedicine journal and e-health : the official journal of the American Telemedicine Association*. doi:https://doi.org/10.1089/tmj.2008.0090

Bal, I. &. (2007). SENSATION - New Nanosensors and Application of Nonlinear Dynamics for Analysis of Biosignals Measured by These Sensors. *World Congress on Medical Physics and Biomedical Engineering 2006* . doi:10.1007/978-3-540-36841-0_172

Bergmann, A. (2014). Benefits and Drawbacks of Model-based Design. *KMUTNB International Journal of Applied Science and Technology*, 15-19. doi:10.14416/j.ijast.2014.04.004

Bhuiyan, M. S. (2009). Driver Assistance Systems to Rate Drowsiness: A Preliminary Study. In Springer, *New Advances in Intelligent Decision Technologies* (pp. pp.415-425). doi:10.1007/978-3-642-00909-9_40

Bishop, C. (2006). *Pattern Recognition and Machine Learning.* Springer. Retrieved February 12th, 2022

Bosch. (n.d.). *Driver Drowsiness Detectinn*. Retrieved February 11th, 2022, from Bosch Mobility Solution: https://www.bosch-mobility-solutions.com/en/solutions/assistance-systems/driver-drowsiness-detection/

Brownlee, J. (2013). *How to Define Your Machine Learning Problem*. Retrieved February 5th, 2022, from Machine Learning Mastery: https://machinelearningmastery.com/how-to-define-your-machine-learning-problem/

Brownlee, J. (2016, March 16th). *Supervised and Unsupervised Machine Learning Algorithms*. Retrieved February 12th, 2022, from Machine Learning Mastery: https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/

Cambridge Coding Academy. (2016). Retrieved from https://cambridgecoding.wordpress.com: https://cambridgecoding.files.wordpress.com/2016/01/knn2.jpg

Ceppi, P. (2020). *Model-based Design of a Line-tracking Algorithm for a Low-cost Mini Drone through Vision-based Control.* MSc thesis. Retrieved February 12th, 2022, from https://webthesis.biblio.polito.it/16018/1/tesi.pdf

Continental AG. (n.d.). *Driver Monitoring System*. Retrieved February 11th, 2022, from Conti-Engineering: https://conti-engineering.com/driver-monitoring-system/

Darrell S. Bowman, W. A. (2012). Advances in Drowsy Driver Assistance Systems Through Data Fusion. In E. A., *Handbook of Intelligent Vehicles.* Springer. doi:https://doi.org/10.1007/978-0-85729-085-4_34

David Sandberg, T. Å. (2011, March). Detecting Driver Sleepiness Using Optimized Nonlinear Combinations of Sleepiness Indicators. *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, 12*(1). doi:10.1109/TITS.2010.2077281

DENSO. (n.d.). *Driver Status Monitor*. Retrieved February 11th, 2022, from www.denso.com: https://www.denso.com/global/en/business/products-and-services/mobility/pick-up/dsm/

Dong, Y. &. (2009). Driver Inattention Monitoring System for Intelligent Vehicles: A Review. In I. V. IEEE (Ed.). IEEE Explore. doi:10.1109/IVS.2009.5164395

Doudou, M. B.-C. (2020). Driver Drowsiness Measurement Technologies: Current Research, Market Solutions, and Challenges. *International Journal of Intelligent Transportation Systems Research*. doi:https://doi.org/10.1007/s13177-019-00199-w

Eleza, R. (2019, October 28th). *AI is the Foundation of Autonomous Vehicles*. Retrieved February 12th, 2022, from iotforall.com: https://www.iotforall.com/autonomous-cars

Eskandarian, A. a. (2007). Evaluation of a Smart Algorithm for Commercial Vehicle Driver Drowsiness Detection. *2007 IEEE Intelligent Vehicles Symposium*. doi:10.1109/IVS.2007.4290173

Eskandarian, A. S. (2007). *Advanced Driver Fatigue Research.* Study summary, U.S. Department of Transportation (USDOT). doi:10.1037/e563992012-001

European Commission. (2011). *White Paper: Roadmap to a Single European Transport Area towards a competitive and resource efficient transport system.* Retrieved February 2, 2022, from http://ec.europa.eu/transport/themes/strategies/2011 white paper en.htm

European Commission. (2019, March 26). *Road safety: Commission welcomes agreement on new EU rules to help save lives.* Retrieved February 11th, 2022, from European Commission.eu: https://ec.europa.eu/commission/presscorner/detail/en/IP_19_1793

European commission. (2021). *Report on Advanced Driver Distraction Warning systems.* Retrieved February 3rd, 2022, from https://ec.europa.eu/docsroom/documents/45901?locale=en

European Transport Safety Council. (2001). The Role of Driver Fatigue in Commercial Road Transport Crashes. Retrieved February 3rd, 2022, from https://etsc.eu/wp-content/uploads/The-role-of-driver-fatigue-in-commercial-road-transport-crashes.pdf

European Transport Safety Council. (2021). *Ranking EU Progress on Road Safety: 15th Road Safety Performance Index Report.* Performance Index Report. Retrieved February 3rd, 2022, from https://etsc.eu/15th-annual-road-safety-performance-index-pin-report/

Eurostat: Statistics explained. (2019, December). *Road safety statistics - characteristics at national and regional level.* Retrieved February 2, 2022, from https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Road_safety_statistics_-_characteristics_at_national_and_regional_level&oldid=463733

Garbade, D. M. (2018, August 11th). *Regression Versus Classification Machine Learning: What's the Difference?* Retrieved February 12th, 2022, from Medium.com: https://medium.com/quick-code/regression-versus-classification-machine-learning-whats-the-difference-345c56dd15f7

Garcia Daza, I. &. (2014). Fusion of Optimized Indicators from Advanced Driver Assistance Systems (ADAS) for Driver Drowsiness Detection. *Sensors*. doi:10.3390/s140101106

Hoddes, E. Z. (1973). Quantification of sleepiness: a new approach. *Psychophysiology*. doi:https://doi.org/10.1111/j.1469-8986.1973.tb00801.x

Ingre, M. A. (2006). Subjective sleepiness, simulated driving performance and blink duration: examining individual differences. *Journal of sleep research*. doi:https://doi.org/10.1111/j.1365-2869.2006.00504.x

Kaida, K. T. (2006). Validation of the Karolinska sleepiness scale against performance and EEG variables. *Clinical neurophysiology : official journal of the International Federation of Clinical Neurophysiology*. doi:https://doi.org/10.1016/j.clinph.2006.03.011

Kecklund, G. &. (1995). Effects of timing of shifts on sleepiness and sleep duration. *Journal of sleep research*. doi:https://doi.org/10.1111/j.1365-2869.1995.tb00226.x

Kumar, D. (2018, December 25th). *Introduction to Data Preprocessing in Machine Learning*. Retrieved January 16th, 2022, from Towards Data Science: https://towardsdatascience.com/introduction-to-data-preprocessing-in-machine-learning-a9fa83a5dc9d

Latysheva, N. (2017). *Implementing Your Own k-Nearest Neighbor Algorithm Using Python*. Retrieved February 21st, 2022, from KDnuggets: https://www.kdnuggets.com/2016/01/implementing-your-own-knn-using-python.html

Li, Z. &. (2017). Performance analysis of K-nearest neighbor, support vector machine, and artificial neural network classifiers for driver drowsiness detection with different road geometries. *International Journal of Distributed Sensor Networks.* doi:10.1177/1550147717733391

Liu, C. C. (2009). Predicting driver drowsiness using vehicle measures: recent insights and future challenges. *Journal of safety research*, 239–245. doi:https://doi.org/10.1016/j.jsr.2009.04.005

Ma, B. (2019, May 16th). *Feature Transformation for Multiple Linear Regression in Python*. Retrieved January 28th, 2022, from Towards Data Science: https://towardsdatascience.com/feature-transformation-for-multiple-linear-regression-in-python-8648ddf070b8

MathWorks. (2019, January 2nd). *What are MIL, SIL, PIL, and HIL, and how do they integrate with the Model-Based Design approach?* Retrieved February 12th, 2022, from MATLAB Central: https://it.mathworks.com/matlabcentral/answers/440277-what-are-mil-sil-pil-and-hil-and-how-do-they-integrate-with-the-model-based-design-approach

MathWorks. (n.d.). *Model based design in Simulink*. Retrieved February 11th, 2022, from MATLAB One: https://matlab1.com/model-based-design-in-simulink/

McCall, J. a. (2005). Lane Change Intent Analysis Using Robust Operators and Sparse Bayesian Learning. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*. doi:10.1109/CVPR.2005.482

McCarthy, J. (2007, November 12th). *jmc.stanford.edu*. Retrieved February 12th, 2022, from http://jmc.stanford.edu/articles/whatisai/whatisai.pdf

McKinsey & Company. (2019). *Start me up: Where mobility investments are going.* Market report. Retrieved February 12th, 2022, from https://www.mckinsey.com/~/media/McKinsey/Industries/Automotive%20and%20Assembly/Our%20Insights/Start%20me%20up%20Where%20mobility%20investments%20are%20going/Start-me-up-where-mobility-investments-are-going.ashx

Mehryar Mohri, A. R. (2012). *Foundations of Machine Learning.* The MIT Press. Retrieved February 12th, 2022

Mitchell, T. M. (1997). *Machine Learning.* New York: McGraw-Hill. Retrieved February 5th, 2022

Patel, H. (2021, August 30th). *What is Feature Engineering — Importance, Tools and Techniques for Machine Learning*. Retrieved February 16th, 2022, from https://towardsdatascience.com/what-is-feature-engineering-importance-tools-and-techniques-for-machine-learning-2080b0269f10

Regan, M. &. (2008). *Driver Distraction: Theory, Effects and Mitigation.* CRC Press. Retrieved February 3rd, 2022, from https://www.researchgate.net/publication/283570098_Driver_Distraction_Theory_Effects_and_Mitigation

Rosario, H. D., Solaz, J., Rodríguez, N., & Bergasa, L. (2010). Controlled inducement and measurement of drowsiness in a driving simulator. *IET Intelligent Transport Systems*. doi:10.1049/iet-its.2009.0110

SAE International. (2021). *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles.* Standard, On-Road Automated Driving (ORAD) committee. doi:https://doi.org/10.4271/J3016_202104

Samiee, S. &. (2014). Data Fusion to Develop a Driver Drowsiness Detection System with Robustness to Signal Loss. Sensors. *Sensors*, 17832-17847. doi:10.3390/s140917832

Sandberg, D. a. (2011). Detecting Driver Sleepiness Using Optimized Nonlinear Combinations of Sleepiness Indicators. *IEEE Transactions on Intelligent Transportation Systems*. doi:10.1109/TITS.2010.2077281

Schmidt, E., Schrauf, M., Simon, M., Fritzsche, M., Buchner, A., & Kincses, W. (2009). Drivers' misjudgement of vigilance state during prolonged monotonous daytime driving. *Accident; analysis and prevention*, 1087–1093. doi:10.1016/j.aap.2009.06.007

Scikit-learn. (n.d.). *Supervised learning*. Retrieved February 12th, 2022, from scikit-learn.org: https://scikit-learn.org/stable/supervised_learning.html

Shen, J. B. (2006). Distinguishing sleepiness and fatigue: focus on definition and measurement. *Sleep medicine reviews, 10(1), 63–76.* doi:https://doi.org/10.1016/j.smrv.2005.05.004

SmartEye. (2020, September 29). *Driver Monitoring (DMS) on its way to becoming mandatory in vehicles around the world*. Retrieved February 11th, 2022, from SmartEye.se: https://smarteye.se/blogs/driver-monitoring-dms-on-its-way-to-become-mandatory-in-vehicles-around-the-world/

Sommerville, I. (2015). *Software Engineering* (X ed.). USA: Pearson. Retrieved February 11th, 2022

Srivastava, T. (2018, March 26). *Introduction to k-Nearest Neighbors: A powerful Machine Learning Algorithm*. Retrieved January 2022, from AnalyticsVidhya:

https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/

Stuart J. Russell, P. N. (2010). *Artificial Intelligence: A Modern Approach.* Prentice Hall. Retrieved February 12th, 2022

The Royal Society for the Prevention of Accidents. (2001). *Driver fatigue and Road accidents: a literature review.* Literature review. Retrieved February 3rd, 2022, from https://www.rospa.com/rospaweb/docs/advice-services/road-safety/drivers/fatigue-litreview.pdf

(2021). *Trends and Future Prospects of the Drowsiness Detection and Estimation Technology.* MDPI. doi: https://doi.org/10.3390/s21237921

Tsamis, A. A. (2021). *Evaluation of the White Paper 'Roadmap to a Single European Transport Area - towards a competitive and resource efficient transport system'.* European Commission, Directorate-General for Mobility and Transport. doi: https://data.europa.eu/doi/10.2832/29547

U.S. Department of Transportation, National Highway Traffic Safety Administration. (2006). *The Impact of Driver Inattention On Near-Crash/Crash Risk: An Analysis Using the 100-Car Naturalistic Driving Study Data.* Retrieved February 3rd, 2022, from https://www.nhtsa.gov/sites/nhtsa.gov/files/810594.pdf

Wakita, T. a. (2005). Driver identification using driving behavior signals. *Proceedings. 2005 IEEE Intelligent Transportation Systems, 2005*. doi:10.1109/ITSC.2005.1520171

Wood, T. (n.d.). *What is Unsupervised Learning?* Retrieved February 12th, 2022, from DeepAI: https://deepai.org/machine-learning-glossary-and-terms/unsupervised-learning

World Health Organization. (2018). *Global Status Report on Road Safety.* Global Status report. Retrieved February 3rd, 2022, from https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries

Wundersitz, L. (2019). Driver distraction and inattention in fatal and injury crashes: Findings from in-depth road crash data. *Traffic injury prevention, 20(7), 696–701.* doi:https://doi.org/10.1080/15389588.2019.1644627