Real-Time Detection System of Driver Distraction Using Machine Learning

(Article begins on next page)

25 April 2024

Real-time Detection System of Driver's Distraction using Machine Learning

Fabio Tango and Marco Botta

Abstract—**There is an accumulating evidence that driver's distraction is a leading cause of vehicle crashes and incidents. In particular, it has become an important and growing safety concern with the increasing use of the so-called In-Vehicle Information Systems (IVIS) and Partially Autonomous Driving Assistance Systems (PADAS). Thereby, the detection of the driver status is of paramount importance, in order to adapt IVIS and PADAS accordingly, so avoiding or mitigating their possible negative effects. The purpose of this paper is to illustrate a method for the non-intrusive and real-time detection of visual distraction, based on vehicle dynamics data and without using the eye-tracker data as inputs to classifiers. Specifically, we present and compare different models, based on well-known Machine Learning methods. Data for training the models were collected using a static driving simulator, with real human subjects performing a specific secondary task (SURT) while driving. Different training methods, model characteristics and feature selection criteria have been compared. Based on our results, SVM has outperformed all the other ML methods, providing the highest classification rate for most of the subjects. Potential applications of this research include the design of adaptive IVIS and of "smarter" PADAS.**

Index Terms — **Accident prevention; artificial intelligence and machine learning; driver' distraction and inattention; intelligent supporting systems.**

I. INTRODUCTION

DRIVER'S inattention and driver's distraction do not have a generally accepted definition: the related terms are frequently discussed in the literature, very often they are inconsistently defined and the relationship between them is unclear [1]-[2]. In addition, neither the extent to which driver distraction is responsible for accidents is completely understood. So, Wang et al., [3] estimated that 13.3% of crashes involve what they considered distraction and 9.7% were in a category called "looked but did not see". Such a percentage can even increase (+2.6%) if drowsiness is considered as well. The 100-Car Naturalistic Driving Study found that almost 80% of all crashes and 65% of all near-crashes involved driver distraction [4]. In fact, it is well-known that the majority of road accidents (surely > 80%) are due to human error [5], or anyway human (wrong) behavior, with an increasing evidence that driver distraction and driver inattention are major contributing factors in car and truck crashes and incidents, with the National Highway Traffic Safety Administration (NHTSA) estimating that, in 25% of all crashes, some form of inattention (including distraction) is involved [6]-[9]. Based on this picture, crashes due to driver distraction, result in as many as 5000 fatalities and $40 billion in damages each year (studies carried out in USA from NHTSA, but also European ones confirm such values, see, e.g., the European projects AIDE and D3COS, http://www.aide-eu.org, or http://www.d3cos.eu/) [10]-[12].

Notwithstanding the ambiguity in its definition and actual impact, it seems that the scientific community agrees on one thing: driver distraction – and inattention – is an important safety concern [13]. All in all, driver's distraction is not a new problem in road safety: we may say that it has been around for as long as people have been driving cars.

It is likely that the problem will increase as more wireless or mobile technologies find their way into vehicles [4], [9], [14]. Although in the last few years many European countries have prohibited the use of – for example – mobile phones when driving, nonetheless it should not be expected that the amount of driving distraction will necessarily decrease. In fact, even without the distraction caused by mobile devices, the use of the so-called In-Vehicle Information Systems (IVIS) – e.g. navigation systems – can be additional sources of potential distraction. One method, followed by many car-manufacturers and automotive suppliers, aims at minimizing the risk of crashes rather than distraction (as pointed out by Wöllmer et al. in [9]) by means of the development of dedicated supporting systems: the so-called Advanced Driving Assistance Systems (ADAS) and Partially Autonomous Driving Assistance Systems (PADAS), such as lane-keeping assistance system, forward collision warning system, emergency braking system, etc. However, it is also true that such PADAS may induce themselves some forms of distraction.

In this context, allowing drivers to take benefits from the use of these IVIS and PADAS without diminishing safety is a big and important challenge. One promising strategy to deal with such a problem involves the classification of driver's status – distracted driver, in this case – in real time and then

Fabio Tango is with Dipartimento E/E Systems, Centro Ricerche Fiat (CRF), Orbassano, TO 10043 Italy (phone: +39-011-9083904; fax: +39-011-9083083; e-mail: fabio.tango@crf.it.

Marco Botta is with Dipartimento di Informatica, Università di Torino, corso Svizzera 185, 10149 Torino, Italy; e-mail: marco.botta@unito.it.

using this classification for a twofold goal: i) the adaption of IVIS technologies, in order to mitigate the effects of distraction, and ii) the adaption of PADAS strategies, in order to minimize the effects of distraction on the driving task.

Machine Learning (ML) and Data Mining (DM) technologies may be able to provide the right algorithms to cope with such a challenge. In fact, ML is the technique of searching large volumes of data for unknown patterns. It has been successfully applied in business, health care and other domains [15]-[16]. In particular, this technology can be applied to build a discrimination model that captures the differences in behavior when people drive normally and when they are distracted.

The main goal of this paper is to present a non-intrusive approach for a real-time system to detect and classify driver's distraction, applying ML algorithms (comparing different methods) and using only vehicle dynamic data as inputs to the model. In particular, here we mainly address the driver visual distraction that has been considered an important aspect in the investigated maneuvers. In this context, looking away for a short while (at least 1.8 seconds) can be considered as a driver visual distraction from her/his main activity.

The paper is organized as follows. Section 2 provides the definition of driver's distraction, based on the current discussion in literature. Then, Section 3 will briefly describe the investigated ML techniques to model driver's distraction. The experimental set-up will be illustrated in Section 4, while Section 5 shows the main results achieved. Finally, Section 6 aims at critically discussing these results, comparing ours to the most important ones obtained by similar works in this area and pointing out the differences, our innovations and weaknesses as well as highlighting possible future activities. Finally, Section 7 concludes the paper with a summary of the main points of interest in this research.

II. DISTRACTION DEFINITION

As mentioned in the introduction, is it possible to reliably detect – and recognize – driver's state, so that the system (such as the PADAS) would give just as much assistance as the driver needs? For instance, the intervention of a forward-collision assistance system can be triggered, based on the driver state: if distraction is detected the function strategies can be adjusted accordingly (e.g. braking is modulated differently or warning signals are anticipated). On the contrary, if the system detects that the driver is not distracted, but intended to overtake, the warning can be delayed or suppressed, even in case of approaching the vehicle ahead. Such a smart assistance, which recognizes driver's intention and state, would allow for a greater safety margin, without irritating the driver with false alarms or inappropriate interventions in normal driving conditions, so enhancing the user acceptability. Therefore, in recent years, several methods have been published, which aims at estimating driver's distraction (for example [17]-[20]) or also which concentrate on the detection and modeling of fatigue or stress as

fundamental causes for driver's inattention (like [21]-[22]).

However, in literature, there is not a unique and commonly agreed definition of distraction, but several ones very often overlapped and mixed with inattention or with other driver's states, such as drowsiness and workload. For what concerns the definition of distraction adopted in our research, we have considered the taxonomy proposed by Regan et al. [2] and by Lee et al. [23]. In particular, we start from the following definition:

"Driver distraction is the diversion of attention away from activities critical for safe driving toward a competing activity".

This has been extended by Regan et al., adding the concept of Driver Inattention, which means insufficient or no attention to critical activities for safe driving toward a competing activity.

It is worth to note that such a definition suffers from hindsight bias, since it is really difficult to say if the driver is distracted until after something dangerous happens and then it will be too late for the system to intervene (Regan mentions this fact in his article). Given that, the same Regan points out that "How to develop taxonomy of driver inattention without the benefit of hindsight is an important theoretical and practical challenge beyond the scope of this paper", so this is still an open-point in literature (and for sure, this is definitely beyond the scope of our paper). Although this statement is absolutely true, nevertheless it would almost be impossible to use the concept of distraction without some preliminary assumptions; even if the situation does not bring to an accident 100 times but it does on the 101st time – even though the behavior is not different – however these are potentially critical situations and we want that our systems can prevent such risky conditions (because we don't know which ones can lead to an accident). In fact, in these situations, drivers are not ready to react appropriately to any unexpected event and thus the accidents are more likely.

To sum up, distinct from other forms of driver inattention, distraction occurs when a driver's attention is diverted away from driving by a secondary task that requires focusing on an object, event, or person not related to the driving task. Although existing data is inadequate and not representative of the driving population, it is estimated that drivers engage in potentially distracting secondary tasks approximately 30% of the time their vehicles are in motion (conversation with passengers is the most frequent secondary task followed by eating, smoking, manipulating controls, reaching inside the vehicle, and cell phone use.). Accordingly to that, we have considered visual distraction as the diversion of visual attention away from the road. This category of "driver distraction" is also the one used by Lee et al. [23].

III. MODELING DRIVER'S STATE

Given the current state of the art and with reference to our previous works (see [19] and [24]) we have selected a widely used ML technique and some other methods not deeply

investigated in literature to model the driver's state: Support Vector Machines (SVM), two types of Neural Networks, static (Feed-Forward Neural Networks FFNN), and dynamic (Layer Recurrent Neural Networks LRNN), and Artificial Neural-based Fuzzy Inference Systems (ANFIS).

A. Description of the SVM Method

Support Vector Machines (SVMs) are arguably one of the most important developments in supervised classification of recent years. Firstly proposed by Vapnik in 1998, SVMs are based on a statistical learning technique and can be used for pattern classification, as well as inference of non-linear relationships between variables [25]-[26]. This method has been successfully applied to a wide variety of domains, such as image processing (e.g. face recognition), text and speech recognition, bioinformatics (e.g. protein classification) [27]. SVMs often achieve superior classification performance compared to other learning algorithms across most domains and tasks; they are fairly insensitive to the curse of dimensionality and are efficient enough to handle very large-scale problems in both sample and variables. The "classical" application of SVMs concerns a binary classification task. The main idea of SVMs is to implicitly map data to a higher dimensional space via a kernel function and then solve an optimization problem to identify the maximum-margin hyper-plane that separates training instances. The hyper-plane is based on a set of boundary training instances, called *support vectors*. New instances are classified according to the side of the hyper-plane they fall into. The optimization problem is most often formulated in a way that allows for non-separable data by penalizing misclassifications.

B. Description of the FFNN Method

Artificial Neural Networks, or simply Neural Networks (ANN or NN), are an information processing system, which is inspired by biological nervous system (the brain) and that consists in a large number of highly interconnected processing elements, working together to solve specific problems [28]. In a neural network, signals are transmitted through connection links, characterized by an associated weight, which is multiplied by the incoming signal (the input of the net) for any typical neural net. The output signal of a unit is obtained by squashing the net input into an activation function. One of the most important types of NNs – used within our research – are the Feed-forward Neural Networks (FFNNs). FFNNs have a layered structure, where each layer consists of units receiving their input from units from a layer directly below and sending their output to units in a layer directly above the unit. There are no connections within units of the same layer. FFNNs are considered static networks, since they have no feedback elements and contain no delays; the output is calculated directly from the input through feed-forward connections.

C. Description of the LRNN Method

Besides the Static (Feed-forward) NNs (whose topology corresponds to acyclic directed graphs), there are also the Dynamic (Recurrent) NNs, where the output depends not only on the current input to the network, but also on the previous inputs, outputs, or states of the network. The Layer-Recurrent Neural Networks (LRNNs), which were introduced by Elman [29] in an earlier simplified version, are a specific type of dynamic networks

All in all, recurrent networks are artificial neural networks that apply to time series data and that use outputs of network units at time t as input to other units at time t+1. Under this viewpoint, they support a form of directed cycles in the network. In the LRNNs, there is a feedback loop, with a single delay, around each layer of the network except for the last layer. In particular, this type of networks are used when the prediction of an output y(t+1) – for example the next day's stock market average, based on the current days economic indicators [30] – depends not only on the input value x(t), but also on earlier values $x(t-i)$, $i \in \{0,1,\ldots,t\}I$.

In order to train a LRNN, a simple variant of the back-propagation method is used. In practice, however, LRNNs are more expensive to train than networks with no feedback loops.

D. Description of the FIS and ANFIS Method

The starting point for talking about Fuzzy Logic (FL) is the consideration about the relative importance of precision: sometimes, a logic based on only two truth values, True and False, can be inadequate when describing human reasoning. FL uses all values inside the interval [0,1] (where 0 is regarded as False and 1 as True) to describe human reasoning and therefore it is a fascinating area of research because it does a good job of trading-off between significance and precision: this is something that humans have been managing for a very long time. In this sense, FL has the ability to mimic the human mind to effectively employ modes of reasoning that are approximate rather than exact.

In more "mathematical terms", Fuzzy Logic is a way to map an input space to an output space. In particular, since a Fuzzy Logic System (FLS) is able to handle simultaneously numerical data and linguistic knowledge, it is a non-linear mapping of an input data (feature) vector into a scalar output (i.e. it maps numbers into numbers). Between the input and output we can put a "black-box" that does the work and inside it, we can find any number of things, from FLS to expert systems, from linear systems to Neural Networks, and so on.

One of the key-concept of FL is the Fuzzy Set(s), which is a set without crisp, clearly defined boundaries. It contains elements with only a partial degree of membership: the truth of any statement becomes a matter of degree. In this context, the Membership Functions (MFs) are curves that define how each point in the input space is mapped to a membership value (also called degree of membership) in the interval [0, 1]. There are several types of MF, but in most cases a Triangular or a Gaussian shape is used.

Fuzzy sets and fuzzy operators can be regarded as the subjects and verbs of fuzzy logic, but in order to say anything useful we need to make complete sentences. So the conditional statements, *if-then rules*, are the things that make fuzzy logic

useful.

Even if the output of each rule is a fuzzy set, in general we want the output for an entire collection of rules to be a single number. To achieve this, firstly the output fuzzy sets for each rule are aggregated into a single output fuzzy set. Then the resulting set is defuzzified, or resolved to a single number. Many defuzzification techniques have been proposed in the literature. Perhaps the most popular defuzzification method is the centroid calculation, which returns the center of area under the curve. For details on FIS, see [31]-[32].

In traditional FIS, the MFs are fixed and somewhat arbitrarily chosen. Moreover, fuzzy inference is applied to modeling systems whose rule structure is essentially predetermined by the user's interpretation of the characteristics of the variables in the model. Hence, one of the key-points in Fuzzy Set Theory is the choice and tuning of membership functions, which are done very often arbitrarily and manually. One possibility is to use the architecture and learning procedure called ANFIS (Adaptive-Network-based Fuzzy Inference System), which is a fuzzy inference system (FIS) in the framework of adaptive networks. An adaptive network is a superset of all kinds of feed forward neural networks with supervised learning capability. In particular, it is a network structure consisting of nodes and direct links through which the nodes are connected. Part or all of the nodes are adaptive, which means that each output of these nodes depends on the parameter(s) pertaining to the node itself; the learning rule specifies how these parameters should be changed to minimize a given error measure. Since the basic learning rule focuses on the gradient method which is notorious for its slowness and tendency to become trapped in local minima, Jang et al. [33] have proposed a hybrid learning rule which can speed up the learning process substantially.

Nowadays, ANFIS has been used in several fields, with well-founded applications to automatic control and signal processing. The nonlinearity and structured knowledge representation of ANFIS are the primary advantages over classical linear approaches in adaptive filtering and adaptive signal processing, such as identification, inverse modeling, predictive coding, adaptive channel equalization, adaptive interference (noise or echo) cancelling, etc. [34]

IV. DESCRIPTION OF THE EXPERIMENTS

The data related to distraction and vehicle dynamic have been collected by means of dedicated experiments using a static driving simulator.

A. Subjects

Twenty participants with a previous experience on the driving simulator have been selected and divided into two groups: ten drivers in the age between 20 and 25 and ten between 30 and 45. A minimum amount of driver experience was required, in particular at least 2 years of driving license and 6000 km driven per year. Driver's gender was not an investigated variable (anyway, there were 3 females and 7 males in each group).

B. Experimental Set-up

As mainly done in other works studying distracted driving, a driving experiment has been conducted on a driving simulator, because of safety issues and better control of the environment, as well as for logistic and economic reasons. In particular, a ScanerII (www.scaner2.com) car simulator has been used: it is a fixed based system that comprises a mock-up of a car with real driving controls (i.e. seat, steering wheel, pedals, gear, handbrake), a digital simulated dashboard displaying a traditional instrumental panel and a frontal projection screen where the simulated environment is displayed to the driver (see Figure 1). Distraction has been induced by means of a secondary visual research task, called SURT (SUrrugate visual-Research Task, a methodology developed by S. Mattes in the project ADAM), reproduced on an in-vehicle display system (7'' TFT touch screen installed on the right-hand side of the car cabin) [50]. Figure 1 shows the situation.

Fig. 1: SURT display on the right part of driving simulator cockpit.

SURT was chosen with the aim of evaluating the interferences caused by a generic visual search task rather than a specific IVIS (In Vehicle Information System), which can be "simulated" in such a way. Like most commercial In-Vehicle Information Systems, it requires visual perception and manual response: such activities, according to Wickens' multiple resources model [35], requires the same mental resources of the driving task and is therefore more likely to interfere, possibly causing a degradation of driving task performances. Of course, each IVIS has a different potential distraction, as a function of its position, size, HMI, working conditions, etc. However, authors believe this is really relevant in real vehicles. Unlike Jiménez et al. investigated in [51], we did not have the possibility to install a real IVIS device in the driving simulator; in the next step of our research, where we will use datasets from real-world, the SURT methodology will be replaced by the use of specific IVIS (such as the navigation system, as done in [9]). All in all, for this type of study based on driving simulator, we think SURT can be effective in modeling the distraction of the drivers.

C. Procedure

Participants performed a practice drive in the driving simulator of 15 minutes. Then, they were asked to drive for

approximately 50 minutes on a simulated three lanes highway: the driving task consisted in keeping the lane and driving at an average speed of 100 km/h at safety distance from the vehicles encountered ahead. For the moment, we have considered a motorway scenario for a couple of reasons: firstly, it represents a more structured and controlled environment; secondly, it is more suitable for the integration with the ADAS application under investigation, the Adaptive Cruise Control (ACC).

During this driving phase, each participant was asked to complete 16 secondary task sessions, each one lasting three minutes. When SURT is activated the display shows a black screen with 30 symbols (each 1.4 cm high), specifically: 14 blue circles, 15 red squares and 1 red circle. The screen is equally divided into two vertical sides and each time the SURT is presented, the driver is asked to touch the side where the red circle is located. The time interval between two consecutive screens was pseudo-randomized between 3 and 9 seconds.

D. Data Collection and Processing

Data of distraction constitute the target set, since we have adopted a supervised learning method. In this methodology using SURT, the eyes-position of subjects has been extracted from videos with a video-processing laboratory software and transferred to a log file as Boolean values (1: eyes on the SURT; 0: eyes on the frontal screen). Then, the change of SURT status, from 0 to 1 and from 1 to 0, has been considered as the key factor to understand if the driver was distracted or not. In fact, from literature ([36]-[37]), if the drivers look away from the road for an interval between 1 and 2 seconds, they can be regarded as distracted. The switches of SURT status identify the period where drivers were engaged with secondary task completion. The number of correct answers together with drivers' reaction time on the SURT (i.e. the difference between the instant the task is presented and the touch of the driver) have been recorded.

Since we consider a supervised learning approach, we needed to define a target-set for the training of the classifiers; this target-set has been built in a post-processing phase, as follows.

An IR camera, with a precision of 25 frames/s, was pointed to the face of the subject and the experimenter visualized on a display if the user was looking at the road or at the SURT. To limit the possible false positives, the experimenter used another camera (same precision), which pointed to the SURT, in order to check if the subject was interacting with it (e.g. click on the SURT display). Every time the experimenter saw that the subject was interacting with the SURT device (so looking away from the road scene), he pressed a "1" on the PC keyboard, otherwise a "0", writing these values into the log-file of the simulator.

Given the log-file, the experimenter has considered all the sequential ones (1) in it, for a period of time equal to at least 1.8s. From literature, this is a good time-period over which a driver can be regarded as visually distracted. When we found this situation, an instance in the target-set was labeled **driver distracted**, otherwise it was labeled **driver not distracted**.

For what concerns the vehicle dynamic data, the following variables have been collected and used:
- Speed [m/s]
- Time To Collision [s]
- Time To Lane Crossing [s]
- Steering Angle [deg]
- Lateral Position [m]
- Position of the accelerator pedal [%]
- Position of the brake pedal [%]

These values are directly available on the prototype vehicle CAN bus (the same one installed on a real vehicle). The frequency of data collection was 20 Hz (1 data-point each 0.05s), which is the output rate of the simulator. Values are then averaged over a period of 1.8s in order to be consistent with the target variable (distracted or not-distracted).

It is worth to note here that these variables constitute the only inputs to the classifiers: the eye-movements data do not appear, since they have been used by the experimenter only to label the target set, as explained before.

Following the ordinary procedure for supervised learning, each data set has been split in three different subsets:
- Training data (around 60% of the whole dataset), which are used to train the classifiers.
- Verification data (around 15% of the whole dataset), which are used to measure classifier generalization and to halt training when generalization stops improving.
- Testing data (around 25% of the whole dataset), which have no effect on training and so provide an independent measure of learning performance after training.

Because of the way the experiment is designed, we consider here the visual distraction (eyes off the road). Although we cannot directly address other types of distraction (e.g. cognitive) by this experiment, nonetheless visual distraction has been shown to be of greatest concern in naturalistic driving studies, as stated in [8] and [56], where both have showed texting (visual distraction) is associated with greater odds to crash-relevant conflict than cell phone conversation (cognitive distraction).

V. DATA ANALYSIS AND RESULTS

In our previous work [24], we had shown an "inter-subject" analysis, where we followed a "leave-one-out" approach: one model has been trained on the data from 9 out of 10 subjects, and tested on the data of the left out subject (in turn, on every subject) and results averaged. Unfortunately, the results were not really satisfactory, since the best obtained performance was around 75% of instances correctly classified (this is a very poor result, meaning that such a classification rate is rarely accepted by users). Very likely, this is due to the fact that the response to distraction is highly personal and subjective, so the normal behavior of one driver can be similar (= too similar for a model) to the distracted behavior of another driver.

Therefore, in this paper we present an "intra-subject" analysis, where one model for each participant is created and the performances of every classifier have been compared, in order to assess how a specific model can fit a specific subject (very interesting for the personalization).

To measure the performances of each classifier, we have considered the following indexes:

- **Correct Rate** (CR), that is the number of instances correctly classified.
- **Sensitivity** (SENS), that is the correctly classified positive instances or True Positive instances.
- **Specificity** (SPEC), that is the correctly classified negative instances or True Negative instances.

In the following, the best model is the one with the highest CR value, a "good model" is a model with CR > 90% and an "acceptable model" is a model with CR > 80% (these values are inferred reading similar works in literature and based on our personal experience).

Finally, we used the MATLAB Neural Networks Toolbox for the FFNN and the LRNN models, the MATLAB Fuzzy Logic Toolbox for the ANFIS model, the MATLAB Bio-informatics Toolbox for the SVM model, and WEKA 3: Data Mining Software in Java for the Logistic Regression model (http://www.cs.waikato.ac.nz/ml/weka/).

The reported values are averaged over 5 runs with the same parameter configuration.

A. Performances for FFNN Distraction Classifier

Different network configurations and topologies have been analyzed for each subject, with different characteristics. The chosen (winning) network has the following characteristic:

- training method = Scaled Conjugate Gradient Back propagation
- number of layers = 2 layers topology has been chosen: one Hidden Layer (HL)); and one Output Layer (OL)
- transfer function = a Sigmoid transfer function has been used for both the HL and OL.

It is very rare the case in which more than one HL is needed; a NN with only a HL can approximate any continuous function. A NN with 2 or more HL can approximate even non continuous functions (in principle, we did not know if the classification function is continuous or not). In the HL, different numbers of Hidden Neurons (HN) have been tested. The Mean Square Error (MSE) has been used to evaluate the performances and as stop-criterion: training automatically stops when generalization stops improving, as indicated by an increase in the MSE on the validation set (that, we remind, is the 15% of the dataset).

As Table I shows, in only 3 out of 20 subjects we obtained a good model, and in other 7 out of 20 subject an acceptable model. The best performance has been obtained for subject 1, with a CR equal to 94.4%. In this case, the training time was 88.2s.

TABLE I
PERFORMANCES OF DIS CLASSIFIER BASED ON FFNN

Subject	HN	CR	MSE
1	**100**	**94,42**	**0,050978**
2	50	79,8	0,149205
3	20	80,42	0,137353
4	50	82,7	0,119556
5	20	84,70	0,115148
6	20	90,32	0,115148
7	50	78,5	0,147561
8	50	85,5	0,109982
9	50	75,8	0,160155
10	50	78,9	0,143094
11	20	79,4	0,138827
12	10	81,4	0,143814
13	10	91,70	0,006357
14	50	79,02	0,142892
15	10	80,2	0,138723
16	50	78,7	0,146673
17	50	74,04	0,165458
18	50	79,02	0,142098
19	10	84,3	0,119444
20	20	76,5	0,156445
Average		81,77	0,121688

HN represents the number of neurons in the hidden layer; *CR* is the correct rate; *MSE* is the mean squared error.

B. Performances for SVM Distraction Classifier

For what concerns SVM, several kernels and different values of their parameters have been tried:

- Linear (LIN)
- Quadratic (QUAD)
- Polynomial (POL)
- Radial Basis Function (RBF)
- Multi-Layer Perceptron (MLP).

The results are reported in Table II.

TABLE II
PERFORMANCES OF DIS CLASSIFIER BASED ON SVM

Subject	Kernel	Parameters	CR
1	RBF	sigma = 0.3	96,89
2	RBF	sigma = 0.2	94,96
3	RBF	sigma = 0,5	96,44
4	POL	order = 5	94,24
5	RBF	sigma = 0.3	94,77
6	RBF	sigma = 0.3	94,92
7	RBF	sigma = 0.3	94,14
8	RBF	sigma = 0,3	96,00
9	RBF	sigma = 0,3	93,79
10	RBF	sigma = 0.4	93,87
11	RBF	sigma = 0,3	93,09
12	RBF	sigma = 0,5	96,62
13	RBF	sigma = 0.2	95,87
14	POL	sigma = 0.3	94,97
15	RBF	sigma = 0.3	94,36
16	**POL**	**order = 5**	**97,95**
17	RBF	sigma = 0.3	97,34
18	RBF	sigma = 0.2	95,05
19	RBF	sigma = 0.3	95,92
20	RBF	sigma = 0.3	94,22
Average			95,27

Kernel is the type of Kernel function used; *Parameters* represent the value of the parameters associated with the specific kernel; *CR* is the correct rate.

As Table II shows, the RBF has proven to be the best Kernel function in 17 cases out of 20. Its expression is:

$$K(x_i, x_j) = e^{-\sigma \|x_i - x_j\|^2}$$

where x_i and x_j represent the data-points and σ is a predefined positive scaling factor parameter.

The RBF is a very robust kernel function, for which it is possible to implement both linear and nonlinear mapping by manipulating the values of its parameters. Moreover, the RBF can reduce numerical difficulties and ends to obtain more robust results than other kernels, such as polynomial and linear (also confirmed by results in [38]).

As shown in Table II, SVM produces good models from every subject. In particular, the best results are obtained for subjects 16 and 17, which have a CR > 97%. The former uses a polynomial kernel, while the latter uses a RBF kernel.

C. Performances for ANFIS Distraction Classifier

In our experiments, we used the Sugeno-type Fuzzy Inference System, which is similar to the Mamdani method in many respects: the first two parts of the fuzzy inference process, input fuzzification and application of the fuzzy operators (see Section III-D), are exactly the same; the main difference between Mamdani and Sugeno is that the Sugeno output membership functions are either linear or constant.

For the creation of this Sugeno-type fuzzy inference systems, there are two main methods for partitioning the input data: the Grid partition and the Sub-clustering. We have selected the second one, which generates an initial model for ANFIS by first applying subtractive clustering to the data. Different values of the clustering centers and radius have been tested, as Table III shows. Best results are achieved with values not exceeding 0.11, in order to have an acceptable trade-off between training time and performances.

The rule extraction method first uses the sub-clustering function to determine the number of rules and antecedent membership functions, then it uses linear least squares estimation to determine each rule's consequent equation. This function returns a FIS structure that contains a set of fuzzy rules to cover the feature space. For this type of FIS structure, Gaussian type membership functions have been used.

In order to optimize membership function parameters, our ANFIS model used a combination of the least-squares method and the back-propagation gradient descent method.

Finally, the training process stops whenever a given number of epochs is reached or the training error goal is achieved.

Table III reports the results obtained by the ANFIS classifier. It achieved very good results, too, producing acceptable models for 7 out of 20 subjects, and good models for the others, with the best performance for subject 1, topping at 96.58% of instances correctly classified.

TABLE III
PERFORMANCES OF DIS CLASSIFIER BASED ON ANFIS

Subject	Radius	CR	Fuzzy Rules
1	**0,1**	**96,5851**	**192**
2	0,08	88,9288	95
3	0,11	89,711	32
4	0,1	93,7912	92
5	0,08	88,15	47
6	0,08	91,0008	43
7	0,08	90,1585	149
8	0,11	90,1278	12
9	0,08	86,9299	65
10	0,08	91,2409	227
11	0,06	91,8674	209
12	0,08	94,1251	141
13	0,08	92,5853	50
14	0,08	91,6188	106
15	0,11	85,0032	24
16	0,08	89,3123	64
17	0,08	91,9695	186
18	0,08	85,7577	85
19	0,08	95,2097	225
20	0,08	92,0561	159
Average		90,8064	

Radius is the value of the radius used for the clustering of data; *CR* is the correct rate; *Fuzzy Rules* represent the number of rules used for the FIS creation.

D. Performances for LRNN Distraction Classifier

All in all, LRNNs are similar to FFNNs, except that each layer has a recurrent connection with a tap delay associated with it. The training function updates weight and bias values according to the BFGS quasi-Newton method (using the algorithm implemented in the *trainbfg* routine of MATLAB). For the adaptation learning function, we have used the gradient descent with momentum weight and bias learning function.

As done with FFNNs, we tried different number of hidden neurons and report results obtained using only one hidden layer (HL), with 20, 50 and 100 neurons. For the HL neurons, we used a tangent-sigmoid transfer function, while for the output layer (OL) neurons we used a linear one.

Also in this case, the MSE error has been used as the performance measure for training the network.

For this type of NN, we only considered 10 subjects, due to the very long training time of the LRNNs (more than 12 hours for each subject) and the poorer results obtained (compared to the other methods). It is worth to note here that LRNNs were an interesting case study, since they have infinite dynamic response and hence time history can be taken into account very easily. The results are reported in Table IV.

TABLE IV
PERFORMANCES OF DIS CLASSIFIER BASED ON LRNN

Subject	Epochs	HN	CR
1	**6**	**100**	**94,42**
2	2	50	79,8
3	6	20	80,42
4	5	50	82,7
5	6	20	84,70
6	6	20	90,32
7	2	50	78,5
8	2	50	85,5
9	2	50	75,8
10	1	50	78,9
		Average	83.11

Epochs is the number of cycles/epochs used by the NN for the training; *HN* represents the number of neurons in the hidden layer; *CR* is the correct rate.

LRNN produced a good model only in 2 cases out of 10, for subjects 1 and 6, with a CR > 90%. The best performances are obtained again for subject 1, and are almost comparable to the other methods.
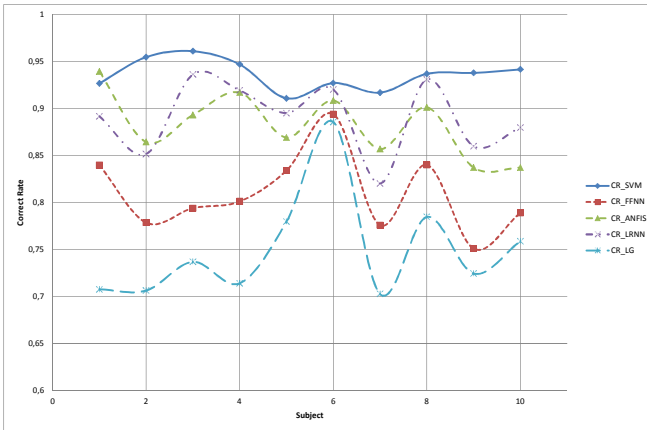

Fig. 2.   Performances of different classifiers (see legend). The measure of performance, the Correct Rate which gives the number of correct instances, is shown for each subject and for each type of classifier.

E.   Comparison between the different Distraction Classifiers

We show here some comparison plots, that report the Correct Rate, Sensitivity and Specificity for every subject, for each classifier. In addition, we report the performance indexes obtained by a **Logistic Regression** (LR) classifier, in order to get a feeling of what we have gained using more complex nonlinear models.

Figure 2 presents the correct rate (CR) of the tested models on the first ten subjects (so LRNN can be included).

The SVM classifier outperforms all the others, even if ANFIS achieves very similar performances. LRNN model provide quite good results, even if not as good as the ANFIS and the SVM. Anyway, it should be pointed out that, LRNNs outperform the FFNN models. The LR classifier presents the worst performances, confirming the fact that using nonlinear models provide valuable gain. This is also confirmed by the sensitivity and specificity plots, as shown in Figures 3 and 4.

From Figure 3, SVM confirms to outperform all the other classifiers in terms of Sensitivity, even if in this case LRNN

and ANFIS obtain very good results. Much worst the behavior of FFNN model, which provides results similar to LR in many cases.

This is also true for the Specificity index in Figure 4, where SVM, ANFIS and LRNN deliver a good capacity to recognize the negative instances as such, while the FFNN and LR give the worst results (LR the least values absolutely).
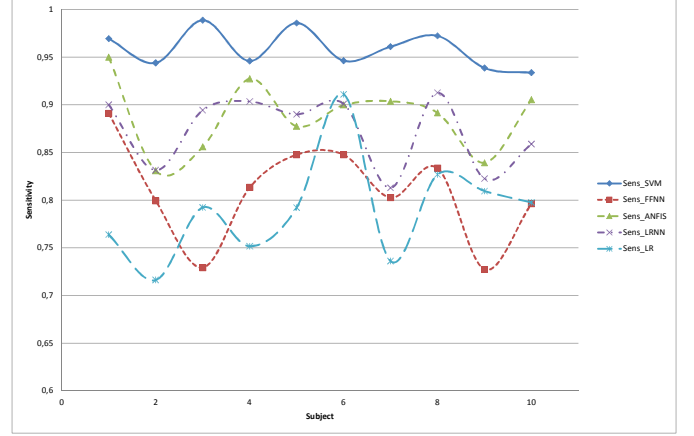

Fig. 3.   Performances of different classifiers (see the legend) concerning the Sensitivity parameter, which measures the proportion of actual positives which are correctly identified as such. This is shown for the first ten subjects, to include LRNN as well.
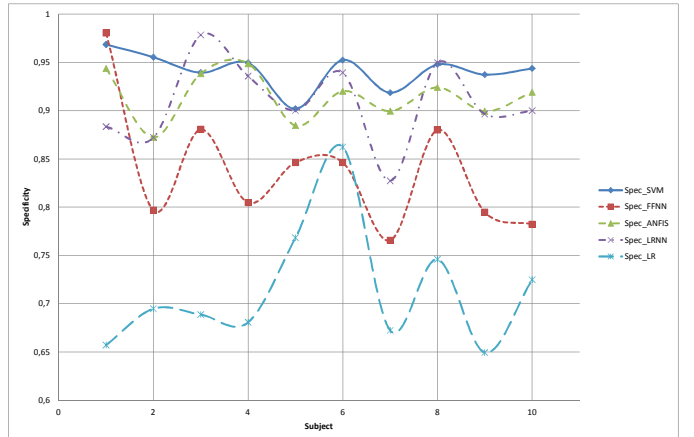

Fig. 4.   Performances of different classifiers (see the legend) concerning the Specificity parameter, which measures the proportion of negatives which are correctly identified. This is shown for the first ten subjects, to include LRNN as well.

It is worth noting that the performances of SVM are quite stable for different subjects, while for the other method they vary quite a lot from subject to subject.

Table V provides a summary of the average performances obtained on all the subjects by the different classifiers, including CR, Sensitivity and Specificity.

TABLE V
AVERAGE PERFORMANCE INDEXES FOR THE DIFFERENT DIS CLASSIFIERS

Type of Classifier	CR	Sens.	Spec.
SVM	**0.95**	**0.96**	**0.95**
ANFIS	0.90	0.89	0.92
LRNN	(0.89)	(0.87)	(0.91)
FFNN	0.81	0.81	0.81
LR	0.75	0.78	0.72

Performance Index (CR), Sensitivity (Sens.) and Specificity (Spec.). LRNN values are computed on the first 10 subjects only (in brackets).

SVM shows the better performances, with respect to all the performance indexes; it is worth to note here that there is a good balance between Sensitivity and Specificity, meaning that the model is able to recognize and classify both positive and negative instances (this is really important, because in real-world, the driver is not distracted most of the time). This tendency can be found also for the other classifiers, even if ANFIS and LRNN show higher values for specificity rather than for sensitivity.

Let's consider now other parameters, in addition to performance. The training time for the best performances of FFNN model took about 99.89s on average for the whole dataset , while the training time of SVM models took 47.06s . Similarly, for LRNN, we have 148408.7s on average and 13804s for ANFIS.

Although SVM training time is the lowest, nonetheless also response time is crucial and under this viewpoint, the NN-based models are usually better. In fact, while SVM has to compute a Kernel function every time, neural networks have an infinitesimal response time, once the weights and the topology have been defined. So, the delay coming from data-reduction and response time of the models, has to be evaluated, since the final choice can depend also on the application.

VI. DISCUSSION

The idea to use ML techniques to detect driver's distraction is not completely new. In particular, [9] and [38] suggest that there are basically three approaches to such a recognition problem: monitoring driver's perception; monitoring driver's steering and lane keeping behavior; recognizing driver's involvement in a given secondary task. Despite the fact that different classification methods can be found in literature to detect distraction or inattention while driving, nevertheless, since the mental state of the driver is not directly observable, no simple measure can weight distraction precisely and thereby all traditional methods show some limits [41]. In this context, the predominant approach is to use ML techniques, which seem to be much more appropriated for this type of classification problem. From a more "philosophical" point of view, one of the most ambitious goals of automatic learning systems is to mimic the learning capability of humans and humans' capability of driving is widely based on experience, particularly on the possibility to learn from experience. From a

more technical point of view, data collected from vehicle dynamics and external environment are definitely non-linear. From literature, several studies have proved that in such situations ML approaches can outperform the traditional analytical methods. Moreover, also human's driver mental and physical behavior is non-deterministic [42]-[45].

On the other hand, vehicle dynamics data are user, road and situation dependent and therefore the classifiers, based on ML techniques, are strongly tailored to the conditions and situation that are selected for the training phase. In fact, we suggest to build a specific model for each driver, and for each situation. How to adapt and generalize such a model to other situations is still an open problem worth to be investigated.

In our opinion, the most representative works are [1], [9], [46], [48], [41] and [24], since more strongly related to our research and they have been a source of inspiration for us.

In particular, the predominant approach is to use static classifiers such as support vector machines (SVMs). Liang et al. developed real-time methods for distraction classification using Support Vector Machines [46] and Bayesian Networks [48]. Their results are comparable to ours, since in [46] they achieved a best performance of more than 95%, while in [47], modeling the dynamic of driver's behavior by using a Dynamic Bayesian Network (DBN), led to accuracies of about 80.1% on average. However, here, the authors pointed out that time dependencies are highly relevant when predicting the current state of a driver. Our best case was > 96%, so – considering also the differences in the experiments, even if both carried out in a driving simulator – absolutely comparable with their best result of 95%. By the way, it is worth noting here that such comparisons can only be indicative, since the datasets are different for each case and also the methods and the tools used for training are not the same.

Similar approaches toward driver behavior or driver state estimation that model contextual information via DBNs or Markov models can also be found in [17] and [49]. Another promising approach can be found in [41], where SVMs are used to detect driver distraction based on data captured under real traffic conditions, resulting in accuracies of 65%–80%. Features are thereby computed from fixed-length time windows, i.e., the amount of context that is incorporated into the classification decision, is predefined.

Other classification strategies include the application of fuzzy logic or neural networks ([34], [37] and [38]).

In addition, it is worth to mention here two specific and recent works: in the former [1], Ersal et al., propose a framework to study the individual effects of secondary tasks and classify driving behavior. They illustrate that the different effects of secondary tasks on different drivers can be studied using a model-based approach. Furthermore, they point out that using the model-based framework in conjunction with SVMs helps systematically classify driving behavior as distracted or non-distracted. In details, this SVM classifier is used with a radial-basis neural-network-based modeling framework, developed to characterize the normal driving behavior of a driver when driving without secondary tasks.

Such a developed model is then used in a scenario of driving with a secondary task to predict the hypothetical actions of the driver: the difference between the predicted normal behavior and the actual distracted behavior gives individual insight into how the secondary tasks affect the driver. When this framework is used together with SVM, it can help systematically classify normal and distracted driving conditions for each driver. So, what is really interesting here is that authors consider a model-based approach, where eye-tracker or gaze data are not present; however, in order to build the target-set, they state: "for the purposes of the classification, all the instances in normal driving are labeled as vigilant and all the instances in driving with secondary task are labeled as distracted". This seems to be inadequate to our scenarios.

In the second [9], Wöllmer et al., introduce a framework and a technique for online driver distraction detection based on modeling contextual information in driving and head tracking data captured during test drives in real traffic. Their approach is based on long short-term memory (LSTM) Recurrent Neural Networks (RNN), exploiting their ability to capture the long-range temporal evolution of data sequences, in order to reliably detect inattention and can be seen as a basis for adaptive lane-keeping assistance. The amount of contextual information that is used for classification is thereby learned by the LSTM network itself during the training phase. This LSTM recurrent neural networks enable a reliable subject-independent detection of inattention with an accuracy > 95%. Thereby, they claim that LSTM framework significantly outperforms conventional approaches such as support vector machines (SVMs).

There are two aspects for which this activity is very interesting for us. Firstly, it is based on data collected on real-car prototype vehicles, while our data are acquired from a driving simulator. Secondly, they address the distraction caused by IVIS (visual and manual), so similar to ours, obtained by SURT. Even though a real comparison on the same data has not been done, we obtained similar results with SVM on the same task, so as they claim LSTM outperforms SVM, it would be interesting to try their approach on our data.

With respect to all these works, our adequate and significant contribution is twofold. The first concerns the comparison of different classification techniques, many of them not considered enough in literature (e.g. ANFIS). The second mainly concerns a different use of input features for the classifiers. In fact, most of the aforementioned works used eye-tracker information as inputs to the classifier. When using the simulator, it is relatively easy to have eye-tracker data, but in a real-time application in the car, this is extremely difficult, since there are several limitations. The first concerns the problem of integration: a dedicated camera and related ECU is needed and has to be integrated into the cockpit of the vehicle (with the associated problems of design and costs). Second, although the information provided by eye-tracker device are absolutely useful, nonetheless they require – for example – that the drivers do not wear sunglasses or glasses, or eye make-up, because these conditions may negatively affect tracking

accuracy [46]. Moreover, there is the problem to obtain consistent and reliable sensor data. Eye trackers may lose tracking accuracy when vehicles are traveling on rough roads or when the lighting conditions are variable. Of course, the use of other physiological measures (such as heart rate or respiration rate, skin conductance, etc.) can provide other excellent indicators, but they are even more intrusive and difficult to use in real-time in the ordinary cars. In this context, our challenge was to provide a data-mining based method, which does not require the mandatory use of eye-tracker information (or other physiological measures) for the classification phase, but it is based only on vehicle dynamic data.

In addition, this research has proved an excellent method to personalize the model; on one side, a "generic" distraction classifier is easier to be extensively applied and trained; however, on the other side, the performances obtained with the application of specific model for each driver are definitely better. Perhaps, this is a direction to take into account in the distraction classification field, since different drivers respond to external or internal stimuli – which are responsible of distraction – in very different manner, as our data proved.

Finally, with respect to our previous research [24], we have extended the analysis both in terms of the ML classification techniques investigated (more models) and in terms of the number of subjects for the experiments (more data-points). In such a way, the results and – above all – the comparisons are much more representative and meaningful.

All in all, some limitations are anyway present in this research. First, with reference to the works of Liang and Wöllmer ([47] and [9]), it would be interesting to explore in more details the approach based on modeling the dynamics of driver behavior, rather than the static network, in order to possibly improve the generalization capability of the classifier itself. So, one of the next steps of our research will involve the use of Dynamic Bayesian Networks and/or of Hidden Markov Models. Furthermore, a deeper investigation of the LRNN classifier will be carried out, due to the very promising results achieved by [9]. In particular, the use of different methods to cluster the data can be considered (e.g. different values for the moving average and also to use the window length as an optimization parameter for the classifier as well), despite the fact that such networks have been extremely long to train. In fact, one key point to consider, as highlighted by [55], consists in taking the time history into account. This has been already done in some way in the pre-processing phase of this research, but as future steps, this information can be included already in the features, in order to benefit all ML approaches.

A second fundamental aspect concerns the need to collect and then to perform tests directly on road data coming from a real prototype car. We have proved that our model can run in real time, but we have assessed it using a driving simulator. In fact, as pointed out by [52] and [53], it is necessary to attempt validation of such research by making comparisons of simulated driving with real road driving. An absolute validation study of driver distractions during real road driving compared to simulated driving would require a comparison of

different levels of distractions, using the road characteristics of the real roads in the simulator and running the same subjects under simulated and real conditions. This activity is foreseen as future research for us and if we could use online data, it would be really interesting to compare our results with those achieved by Wöllmer et al. In addition, as mentioned in [54], Greenberg et al. indicated that motion cueing may have a strong impact on lateral driving PIs when disturbances (secondary tasks) are present. This point has to be consider as well, in order to verify if the features we used are meaningful in real driving situations. Currently, we can say that other works, such as [9], have already considered and used them.

Finally, we want to test our distraction classifier in a more diverse set of conditions and scenarios (in simulator or real-traffic); in this work we have mainly investigated the motorway, but we want to extend the experimental phase in urban scenarios above all, which is a fundamental step in order to assess the generality of our results.

VII. CONCLUSIONS

This paper presented an overview of different driver's distraction classifiers based on ML techniques. We explored the performances of several models: SVM, FFNN, LRNN and ANFIS. All have been proved to constitute a viable means of detecting driver's inattention, whose cognitive and visual distractions are particular forms. In the current research we pointed out the personalization aspect, with one specific model for each subject. With reference to the results illustrated in Section V, SVM outperformed all the other classifiers, for which we have obtained an accuracy comparable to the one in literature. Our major innovative aspect consists in not using eye-movements or head-movements information as inputs for the classifier.

The European co-funded Integrated Project D3COS (http://www.d3cos.eu/), started in March 2011, allows us to investigate at least some of the future activities mentioned in the previous section.

REFERENCES

[1] T. Ersal, H.J.A. Fuller, O. Tsimhoni, J.L: Stein and H.K. Fathy. "Model-based Analysis and Classification of Driver Distraction under Secondary Tasks" in IEEE Transaction on Intelligent Transportation System , vol. 11,No. 3, September 2010, IEEE Society Editor.

[2] M.A. Regan, C. Hallet and C.P. Gordon. "Driver Distraction and Driver Inattention: definition, relationship and taxonomy". In Accident Analysis and Prevention journal, Elsevier, vol. 43, 2011.

[3] C J.-S. Wang, R. R. Knipling, and M. J. Goodman, "The role of driver inattention in crashes: New statistics from the 1995 crashworthiness data system," in Proc. 40th AAAM, Vancouver, BC, Canada, 1996, pp. 377–392.

[4] T. A. Dingus, S. G. Klauer, V. L. Neale, A. Petersen, S. E. Lee, J. Sudweeks, M. A. Perez, J. Hankey, D. Ramsey, S. Gupta, C. Bucher, Z. R. Doerzaph, J. Jermeland, and R. R. Knipling, "The 100-car naturalistic driving study, phase II-results of the 100-car field experiment," Nat. Highway Traffic Safety Admin., Washington, DC, Dept. Transp., HS 810 593, 2006.

[5] J. Treat et al. "Tri-level Study of the Causes of Traffic Accidents: Final Report". volume 1. Technical Report Federal Highway Administration, US DOT (1979).

[6] Yanchao Dong, Zhencheng Hu, Member, IEEE, Keiichi Uchimura, and Nobuki Murayama, Driver Inattention Monitoring System for Intelligent Vehicles: A Review, IEEE Transactions on Intelligent Transportation Systems, Vol 12, nº2, June 2011

[7] Thomas A. Ranney, ?Driver distraction: a review of the current state-of-knowledge?. NHTSA. DOT HS 810 704. April 2008.

[8] R.L.. Olson, R.J. Hanowski, J.S. Hickman, J. Bocanegr. "Driver distraction in commercial vehicle operations". Report No. FMCSA-RRR-09-042, U.S. Department of Transportation, Washington, D.C., 2009.

[9] M. Woeller, C. Blaschke, T. Schhindl, B. Schuller, B. Faerber, S. Mayer and B. Trefflich. "Online Driver Distraction Detection using long short-term Memory" in IEEE Transaction on Intelligent Transportation System , vol. 12,No. 2, June 2011, IEEE Society Editor.

[10] E.D. Sussman, H. Bishop, B. Madnick and R. Walter. "Driver inattention and highway safety," Transp. Res. Rec., no. 1047, pp. 40–48, 1985.

[11] J. Wang, R.R. Knipling and M. J. Goodman. "The role of driver inattention in crashes; New statistics from the 1995 crashworthiness data system (CDS)," in Proc. 40th Annual Association Advancement Automotive Med., 1996, pp. 377–392.

[12] J.C. Stutts, D W. Reinfurt, L: Staplin and E.A. Rodgman. "The role of driver distraction in traffic crashes," AAA Foundation of Traffic Safety, 2001.

[13] Caird, J.K., Dewar, R.E., 2007. "Driver distraction". In Dewar, R.E., Olsen, R. (Eds.), Human Factors in Traffic Safety. , 2nd ed. Lawyers & Judges Publishing, Tucson, AZ, USA, pp. 195–229.

[14] D. Haigney and S. J. Westerman. "Mobile (cellular) phone use and driving: A critical review of research methodology" in Ergonom., vol. 44, no. 2, pp. 132–143, Feb. 2001.

[15] P.-N. Tan. "Introduction to Data Mining. Pearson". Addison Wesley, Boston, 2005.

[16] P. Baldi and S. Brunak. "Bioinformatics: The Machine Learning Approach". 2nd edition ed. MIT Press, 2001.

[17] A. Pentland and A. Liu, "Modeling and prediction of human behavior," Neural Comput., vol. 11, no. 1, pp. 229–242, Jan. 1999.

[18] L. Qiao, M. Sato, and H. Takeda, "Learning algorithm of environmental recognition in driving vehicle," IEEE Trans. Syst., Man, Cybern., vol. 25, no. 6, pp. 917–925, Jun. 1995.

[19] Tango F., Botta M. Evaluation of Distraction in a Driver-Vehicle-Environment Framework: an application of different Data-mining techniques. In Proc. 9th Industrial Conference on Data Mining (ICDM09). Leipzig, Germany, 2009. Springer-Verlag Eds.

[20] C. Blaschke, F. Breyer, B. Färber, J. Freyer, and R. Limbacher, "Driver distraction based lane-keeping assistance," Transp. Res. Part F: Traffic Psychol. Behav., vol. 12, no. 4, pp. 288–299, Jul. 20

[21] H. Zhang, M. R. H. Smith, and G. J. Witt, "Identification of real-time diagnostic measures of visual distraction with an automatic eye-tracking system," Hum. Factors, vol. 48, no. 4, pp. 805–821, 2006

[22] J. Hoel, M. Jaffard, P. Van Elslande, 2010. Attentional competition between tasks and its implications. Paper presented at the European Conference on Human Centred Design for Intelligent Transport Systems (29–30 April). Retrieved from http://www.conference2010.humanist-vce.eu/.

[23] J.D. Lee, K.L. Young, M.A. Regan, 2008. Defining driver distraction. In: Regan, M.A., Lee, J.D., Young, K.L. (Eds.), Driver Distraction: Theory, Effects, and Mitigation. CRC Press Taylor & Francis Group, Boca Raton, FL, USA, pp. 31–40

[24] Tango, F., Minin L., Montanari, R., & Botta, M. (2010, August, 16-19). Non-intrusive Detection of Driver Distraction using Machine Learning Algorithms. In the proceeding of the XIX European Conference on Artificial Intelligence (ECAI). Lisbon, Portugal (ISBN: 978-1-60750-605-8; Editor: IOS Press Amsterdam, The Netherlands, The Netherlands).

[25] Vapnik V. N., The Nature of Statistical Learning Theory. New York: Springer-Verlag, 1995.

[26] Cristianini N. and Taylor J. S., An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge, U.K.: Cambridge Univ. Press, 2000.

[27] Byun H. and Lee S. W., "Applications of Support Vector Machines for pattern recognition: A survey," in Proc. 1st Int. Workshop, SVM Pattern Recog. With Support Vector Mach., Niagara Falls, ON, Canada, 2002, pp. 213–236.

[28] Haykin S.: Neural Networks: a comprehensive Foundation. Prentice Hall (1999).

[29] Elman, J.L. (1990). "Finding structure in time," Cognitive Science, Vol. 14, pp. 179–211.

[30] Mitchell; M. (1997). "Machine Learning"; Mc-Graw Hill International Editions.

[31] Bezdec, J.C. (1981). Pattern Recognition with, Fuzzy Objective Function Algorithms, Plenum Press, New York.

[32] A. Kaufmann and M.M. Gupta. Introduction to Fuzzy Arithmetic, V.N. Reinhold, 1985.

[33] Shing J., Jang, R. (1993). ANFIS: Adaptive-Network-Based Fuzzy Inference System. IEEE Transactions on Systems, Man and Cybernetics, 23 (3): 665- 685D..

[34] Sugeno M. Industrial applications of fuzzy control, Elsevier Science Pub. Co., 1985.

[35] Wickens C.D. Multiple Resources and performance prediction. Theoretical Issues In Ergonomics Science, 3(2), 159-17 (2002).

[36] Mattes, S. (2003). The lane change task as a tool for driver distraction evaluation. In H. Strasser, H. Rausch & H. Bubb (Eds.), Quality of work and products in enterprises of the future, Stuttgart: Ergonomia Verlag, pp. 57-60.

[37] Young K. Regan M. and Hammer M. Driver Distraction a review of the Literature. Technical Report No 206, Monash University, accident Research Center, November 2003.

[38] Y. Zhang, Y. Owechko, and J. Zhang. Driver cognitive workload estimation: A data-driven perspective. In Proc. IEEE Intell. Transp. Syst. Conf., Washington, DC, 2004, pp. 642–64

[39] F. Tango, R. Aras and O. Pietquin (09 – 14 July, 2011) Automation Effects on Driver's Behaviour when integrating a PADAS and a Distraction Classifier. Human Computer Interaction Conference (HCI2011). Orlando, Florida (USA).

[40] F. Tango, R. Aras and O. Pietquin (2010, December, 10-11). Batch reinforcement learning for optimizing driving assistance strategies. In Neural Information Processing Systems (NIPS) 2010 conference. Vancouver, Canada.

[41] M. H. Kutila, M. Jokela, T. Mäkinen, J. Viitanen, G. Markkula, and T. W. Victor, "Driver cognitive distraction detection: Feature estimation and implementation," Proc. Inst. Mech. Eng., Part D: J. Automobile Eng., vol. 221, no. 9, pp. 1027–1040, 2007.

[42] D. de Waard, K. A. Brookhuis, and N. Hernandez-Gress, "The feasibility of detecting phone-use related driver distraction," Int. J. Vehicle Des., vol. 26, no. 1, pp. 85–95, 2001.

[43] T. D'Orazio, M. Leo, C. Guaragnella, and A. Distante, "A visual approach for driver inattention detection," Pattern Recognit., vol. 40, no. 8, pp. 2341–2355, Aug. 2007.

[44] Q. Ji, P. Lan, and C. Looney, "A probabilistic framework for modeling and real-time monitoring human fatigue," IEEE Trans. Syst., Man, Cybern. A, Syst., Humans, vol. 36, no. 5, pp. 862–875, Sep. 2006.

[45] J. A. Healey and R.W. Picard, "Detecting stress during real-world driving tasks using physiological sensors," IEEE Trans. Intelligent Transp. Syst., vol. 6, no. 2, pp. 156–166, Jun. 2005.

[46] Y. Liang, M. L. Reyes, and J. D. Lee, "Real-time detection of driver cognitive distraction using support vector machines," IEEE Trans. Intelligent Transp. Syst., vol. 8, no. 2, pp. 340–350, Jun. 2007.

[47] Y. Liang, J. D. Lee, and M. L. Reyes, "Nonintrusive detection of driver cognitive distraction in real time using Bayesian networks," Transp. Res. Rec.: J. Transp. Res. Board, vol. 2018, pp. 1–8, 2007.

[48] Y. Liang and J. D. Lee, Driver Cognitive Distraction Detection Using Eye Movements. Berlin, Germany: Springer-Verlag, 2008, pp. 285–300.

[49] T. Kumagai and M. Akamatsu, "Prediction of human driving behavior using dynamic Bayesian networks," IEICE Trans. Inf. Syst., vol. E89-D, no. 2, pp. 857–860, Feb. 2006.

[50] L. Minin et al. Report on the influence of relevant factors (characteristics, state, situation) on driving behavior and driving errors. Internal Project Deliverable – ISI-PADAS EU project (2009).

[51] P. Jiménez, L.M. Bergasa, J. Nuevo, I. G. Daza, N. Hernández, Gaze Fixation System for the evaluation of Driver Distractions induced by IVIS, IEEE Transactions on Intelligent Transportation Systems, (2012). [On line]. Doi: 10.1109/TITS.2012.2187517.

[52] David Sandberg, "Detecting Driver Sleepiness", PhD. Department of Applied Mechanics. Chalmers University of Technology. Göteborg, Sweden (2011)

[53] Bergasa, L.M., Nuevo, J., Sotelo, M.A., Barea, R., Lopez, E. "Real-time system for monitoring driver vigilance". Intelligent Transportation Systems, IEEE Transactions on 7(1), 63?77 (2006)

[54] Greenberg, J., Artz, B., and Cathey, L. (2003). The Effect of Lateral Motion Cues During Simulated Driving, DSC North America 2002 Proceedings, Dearborn, MI (CDROM).

[55] Green, P. E., Wada, T., Oberholtzer, J., Green, P. A., Schweitzer, J., & Eoh, H. (2007). How do distracted and normal driving differ: An analysis of the ACAS naturalistic driving data (Technical Report No. UMTRI-2006-35). Ann Arbor, MI: The University of Michigan Transportation Research Institute.

[56] Hickman, J. S., Hanowski, R.J., & Bocanegra, J. (2010). Distraction in Commercial Trucks and Buses: Assessing Prevalence and Risk in Conjunction with Crashes and Near-Crashes (FMCSA-RRR-10-049). Washington, DC: U.S. Department of Transportation.

Fabio Tango received the degree in Physics (Solid State) from the University of Turin, Italy, in 1995 and the PhD in Computer Science from University of Turin in 2009.

Currently, he is senior researcher at the Fiat Center of Research (CRF) since 1999, in the "Electric and Electronic Systems" department, where he is working in the area of preventive safety systems. In particular, he is the technical coordinator of the projects ISI-PADAS and D3COS, as well as the CRF responsible of INTERACTIVE / PERCEPTION integrated project (all co-funded by EC).

His topics of interest include: human-automation interaction; user's status modeling (distraction and fatigue detection); pattern recognition and analysis; data-fusion and objects classification, with applications in image processing; applications of ML techniques for automatic driving.

Marco Botta graduated, with laude, in Computer Science at Universita` di Torino in 1987. He received the Ph. D. in Computer Science from Universita` di Torino in 1993.

Since Nov. 2001, he is an associate professor of Computer Science at the Faculty of Mathematical, Physical and Natural Sciences of the Universita` di Torino, Italy. He is affiliated to the Dipartimento di Informatica of the same University.

His research activity was mainly focused on artificial intelligence topics and, in particular, on machine learning problems. In his early research years he mainly studied and developed new methods for learning concepts from instances. Such methods were oriented both to the construction of knowledge bases for expert systems and to their refinement. Since 1996 his research activity has been devoted to the integration of symbolic and sub-symbolic learning approaches with the aim of combining the expressive power of first-order logic with the refinement mechanisms that are typical of a connectionist approach. Since 2002 he is also interested in bioinformatics research and has developed many collaborations with biologists on several topics, such as protein folding prediction, microarray data analysis, and more recently transcription factor binding sites prediction.