On-line Driver Distraction Detection using Long Short-Term Memory

Martin Wöllmer*, Christoph Blaschke, Thomas Schindl, Björn Schuller,
Berthold Färber, Stefan Mayer, Benjamin Trefflich

***Abstract*—Lane-keeping assistance systems for vehicles may be more acceptable to users if the assistance was adaptive to the driver's state. To adapt systems in this way, a method for detection of driver distraction is needed. Thus, we propose a novel technique for on-line detection of driver's distraction, modeling the long-range temporal context of driving and head tracking data. We show that Long Short-Term Memory (LSTM) recurrent neural networks enable a reliable, subject-independent detection of inattention with an accuracy of up to 96.6 %. Thereby our LSTM framework significantly outperforms conventional approaches such as Support Vector Machines.**

***Index Terms*—Driver assistance systems, Driver state estimation, Recurrent Neural Networks, Long Short-Term Memory.**

I. Introduction

DRIVER inattention is one of the major factors in traffic accidents. The National Highway Traffic Safety Administration estimates that in 25 % of all crashes some form of inattention is involved [1]. Distraction (besides drowsiness), as one form of driver inattention, may be characterized as: "any activity that takes a driver's attention away from the task of driving" [2]. Causes for driver inattention are for example the use of wireless devices or passenger related distractions [3]. Although over the last few years many European countries have prohibited, for instance, the use of wireless devices while driving, it should not be expected that the amount of distraction in driving will necessarily decrease. Even without the distractions caused by mobile devices, the amount of distraction due to in-car information systems will increase. Thus, original equipment manufacturers (OEMs) and automotive suppliers will need to find a way to deal with this problem.

One method that minimizes crashes rather than distractions is the development of new driver assistant systems [4], [5]. With the evolution of adequate lane tracking, lane keeping assistance systems were introduced into the market recently. These systems track the lane markings in front of the vehicle and compute the time until the vehicle will cross the marking. If the driver does not show an intention of leaving the lane by using the indicator, the systems will use directed steering torques on the steering wheel to guide the car to the middle of the lane. Authors of several studies reported overall effects of lane departure warning systems on lane keeping performance [6]–[8]. Even though different kinds of warnings can be helpful, participants in [7] judged the lane departure warning system to be annoying in some circumstances. The reason why those systems are annoying for some drivers is easy to explain. That is, lane keeping assistance aims at preventing the driver from making unintended lane departures. However, these systems do not yet respond to the driver's state or his intent but to lane markings and the car's speed. This implies that warnings can be triggered if attentive drivers intentionally change lanes but forget to use the indicator or if certain maneuvers that are executed with full attention require lane crossings. Thus, if it was possible to recognize a driver's state reliably, the system would give just as much assistance as the driver needed. This would allow for a greater safety margin without irritating the driver with false alarms in normal driving situations.

In [9] three main approaches to such a recognition are discussed: monitoring of driver's perception, monitoring of driver steering and lane-keeping behavior, and the recognition of the driver's involvement in a secondary task itself. In recent years, several techniques trying to estimate the driver to be distracted have been published. However, the majority of approaches are developed and evaluated using data that was captured in a driving simulator and not in a real vehicle, where data is much more noisy and complex than it is in a simulator scenario [10]–[14]. A considerable number of studies concentrate on the detection and modeling of fatigue as an important cause for inattention (e. g. [15]–[17]). However, as shown in [12], also visual distraction downgrades driving performance.

In order to detect distraction or inattention while driving, different classification techniques can be found in literature. The predominant approach is to use static classifiers such as Support Vector Machines (SVM) [13], [18]. A promising approach can be found in [19] were SVM are used to detect driver distraction based on data captured in 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. In [14], the authors show that time-dependencies are highly relevant when predicting the current state of a driver: modeling the *dynamics* of driver behavior by using a Dynamic Bayesian Network (DBN) rather than a static network led to accuracies of around 80 %. Similar approaches towards driver behavior or driver state estimation that model

M. Wöllmer, T. Schindl, and B. Schuller are with the Institute of Human-Machine-Communication, Technische Universität München, Theresienstr. 90, 80333 München, Germany (e-mail: {woellmer,schuller}@tum.de, tel.: +49-89-28928550, fax.: +49-89-28928535)

C. Blaschke and B. Färber are with the Human Factors Institute, Universität der Bundeswehr München, Germany

S. Mayer and B. Trefflich are with the Audi Electronics Venture GmbH, Germany

Manuscript received month day, year; revised month day, year.

contextual information via DBNs or Markov models can also be found in [20] and [21]. Other popular classification strategies include the application of fuzzy logic [22], multiple adaptive regression trees [10], or neural networks [11], [16].

Neural networks are able to model a certain amount of context by using cyclic connections. These so-called recurrent neural networks (RNN) can in principle map from the entire *history* of previous inputs to each output. Yet, the analysis of the error flow in conventional recurrent neural nets led to the finding that long-range context is inaccessible to standard RNNs since the backpropagated error either blows up or decays over time (vanishing gradient problem [23]). This led to the introduction of Long Short-Term Memory (LSTM) RNNs [24]. They are able to overcome the vanishing gradient problem by using memory cells to store and access information over long time periods and can learn the optimal amount of contextual information relevant for the classification task – a property that is highly beneficial for predicting the state of a driver.

In this contribution we introduce a framework for online driver distraction detection based on modeling contextual information in driving and head tracking data captured during test drives in real traffic. Our approach is based on Long Short-Term Memory RNNs, exploiting their ability to capture the long-range temporal evolution of data sequences. We investigate both, 'sample-wise' prediction based on low-level signals and 'frame-wise' classification using statistical functionals of the signals. We demonstrate that using low-level signals for driver distraction detection is hardly feasible with conventional recurrent neural networks where the amount of accessible context information is limited.

This article is structured as follows: Section II introduces the accomplished test drives in real traffic and the resulting database that has been used for training and evaluating our driver distraction detection system, Section III provides an overview over the architecture of our system, Section IV outlines the signal pre-processing and feature extraction we used, Section V briefly reviews the basic principle of Long Short-Term Memory while Section VI shows experimental results. Conclusions are drawn in Section VII.

II. Database and Signals

In order to collect data that represents a distracted drivers' behavior in realistic driving situations, 30 participants (12 female and 18 male) were recruited. The subjects were 23 to 59 years old and had driven at least 10.000 kilometres in the last 12 months. An Audi A6 was used as the experimental car. The car was equipped with the Audi Multimedia System (see Figure 1) and an interface to measure Controller Area Network (CAN)-Bus data. Additionally, a head tracking system [9] was installed, which was able to measure head position and head rotation. This data was also sent on CAN-Bus. Head tracking systems are not common in vehicles today, but promising research in systems for driver state detection will lead to a higher installation rate in serial cars in the near future. So we decided to use head tracking information in our approach as well.

Fig. 1. Audi A6 Cockpit

Eight typical tasks on the Multimedia Interface were chosen as distraction conditions:

- radio: adjust the radio sound settings (choose the submenu 'sound', adjust treble and bass to the middle position, return to the 'radio' menu)
- CD: skip to a specific song (search for the song 'sail away' and select it; CD already inserted)
- phonebook: search for a name in the phonebook (find the name 'Werner Blaschke', make a call and hang up immediately)
- navigation-point of interest: search for a nearby gas station (find the nearest 'ESSO' gas station, start route guidance)
- phone: dial a specific phone number (manually dial a number consisting of eleven digits)
- navigation: enter a city in the navigation device (manually enter 'Burgholzhausen-Center', start route guidance)
- TV: switch the TV mode to 'PAL' (change TV-norm from North American to European)
- navigation-sound: adjust the volume of navigation announcements (adjustment to medium volume)

We exclusively focused on these kinds of visual and manual distractions that are typical when operating in-vehicle information systems. Purely mental forms of distraction or inattention (such as 'being lost in thought') were excluded since they are comparably hard to elicit and detect. Also tasks leading to auditory distraction (e. g. talking to a passenger) were not included in our experiments as they are generally considered as low-risk activities [25].

The main functions (e. g. navigation, CD/TV, and radio) are available through eight so-called hardkeys which are located on the right- and left-hand side of the control button (see Figure 1). In each main menu, special functions (e. g. sound settings in the radio menu) can be selected by the four so-called softkeys which surround the control button. These special functions differ between the main menus. The functions assigned to the softkeys are shown in the corners of the display which is located in the middle console.

Most inputs are done using the control button. By turning the control button left or right it is possible to scroll up and down in lists while pushing the button selects highlighted items. For typing letters (navigation) or digits (phone) the so-called speller is used, whereas symbols are arranged in a circle

and can be selected by turning and pushing the control button.

As an example, the following steps have to be done in order to enter a city in the navigation device:

- press the hardkey 'NAV'
- select 'Enter Destination' in a list (one row down)
- use the speller nine times to enter the city
- press the control button to confirm the city
- select 'Downtown' in a list (one row down)
- confirm 'Start Navigation'

The procedure for the experiment was as follows: after a training to become familiar with the car each participant drove down the same country road eight times (one time per task) while performing secondary tasks on the in-vehicle information system. Each task was performed only once per drive and only the time from the beginning of the task to the end of the task was recorded as a 'distracted drive'. On another two runs the drivers had to drive down the road with full attention on the roadway ('baseline' runs). In order to account for sequential effects, the order in which the conditions were presented was randomized for each participant. During each drive CAN-Bus data (including head tracking data) were logged.

The experiments were performed on a German country road with an average road width of 3.37 m and continuous road marking (Ayingerstr. between Faistenhaar and Aying, Bavaria). The route has no sharp turns and consists of one lane per direction. During the experiments oncoming traffic was present, however, the overall traffic density was moderate. Participants drove during the daytime under different weather conditions (mostly dry).

Overall, 53 runs while driving attentively and 220 runs while the drivers were distracted could be measured (some runs had to be excluded due to logging problems). The 'attentive' runs lasted 3 134.6 seconds altogether, while 9 145.8 seconds of 'distracted' driving were logged. Thus, the average duration of attentive and distracted runs was 59.2 seconds and 41.6 seconds, respectively. At an average speed of roughly 100 km/h, this corresponds to distances of 1.64 km and 1.16 km, respectively.

An analysis of the influence on lane keeping of the different in-vehicle information system interaction tasks [9] indicated that the tasks can be characterized as distracting in general.

As will be explained in Section VI, we consider three different classification tasks for the estimation of distraction: the binary decision whether a driver is distracted or not (two-class problem), the discrimination between no, medium, and a high degree of distraction (three-class problem), and the discrimination between six levels of distraction (six-class problem). For the binary problem examined in Section VI, all tasks (i. e. runs during which the tasks were performed) were labeled as 'distracted' compared to driving down the road with full attention ('attentive'). Since all participants were asked to judge the level of distraction of a certain task (meaning the difficulty of the task) on a scale between 1 (easy) and 5 (difficult), these individual judgments were used to model also the *degree* of distraction as a six-class problem ('attentive' plus five levels of distraction, see Section VI). For the three-class problem, difficulties 1 to 3 as well as difficulties 4 and 5 were clustered together. Thus, our system for driver distraction detection is trained to predict the subjective ratings of distraction assigned by the participants using different levels of granularity. Even though the system outputs a prediction for the level of distraction every few milliseconds, the level of distraction is defined *by drive*, meaning that we assign the same level of distraction to each time step of a certain drive. This has the effect that the classifier considers long-term context and predicts the driver state according to the overall difficulty of the task and the resulting level of distraction. We assume that during the 'distracted' runs the driver is continuously engaged in the task, even if there are short periods of attention which are of course necessary while driving. By characterizing distraction on a *per-drive* basis, we smooth out these short intervals of attention in order to model the driver state on a long-term basis which in turn is desired when using driver state predictions for adaptive lane keeping assistance.

Six signals were chosen for a first analysis:

- steering wheel angle (SA)
- throttle position (TP)
- speed (SP)
- heading angle (HA, angle between the longitudinal axis of the vehicle and the tangent on the center line of the street)
- lateral deviation (LD, deviation of the center of the car from the middle of the traffic lane)
- head rotation (HR, rotation around the vertical axis of the car)

The first three (SA, TP, and SP) are direct indicators of the driver behavior. Many studies prove the fact that visually distracted drivers steer their car in a different way than attentive drivers do. The same applies for throttle use and speed (an overview can be found in [25]). The car's heading angle and its lateral deviation in the lane rely on the amount of attention the driver is allocating to the roadway and may hence give useful information about distraction. Head rotation of the driver is an indicator of the driver's visual focus. While using the Multimedia Interface, which is located in the middle console just below the dashboard, the main rotation of the head is to the right. Thus, the heading angle of head rotation is the most promising indicator of the head tracking signals.

III. System Overview

The main architecture of our system for driver distraction classification can be seen in Figure 2. In the following we will denote all signals prior to statistical functional computation as *low-level signals* with synchronized time index t (and time index t' prior to synchronization) whereas f is the frame index referring to the time windows over which statistical functionals are calculated. In Section VI we will investigate both, the direct modeling of low-level signals $s(t)$ (including the first and second derivatives) and the modeling of statistical functionals of those signals ($x(f)$). In other words, we examine the performance of driver distraction detection with and without the processing unit represented by the dotted box in Figure 2. Thereby statistical functionals can be parameters such as extremes, percentiles, means, etc. (see Section IV).

A camera capturing the road in front of the vehicle provides a video signal $v_1(t')$ which is processed by the lane departure warning system to compute the current lateral deviation $s_{LD}(t')$ and heading angle $s_{HA}(t')$. The head rotation $s_{HR}(t')$ is determined by a head tracking system that processes the signal $v_2(t')$ recorded by a second camera facing the driver. Steering wheel angle $s_{SA}(t')$, throttle position $s_{TP}(t')$, and speed $s_{SP}(t')$ are captured by the corresponding sensors and sent to the CAN-Bus together with $s_{LD}(t')$, $s_{HA}(t')$, and $s_{HR}(t')$.

The sample frequencies of the six signals represented by $s_c(t')$ range from 10 to 100 Hz. Thus, the data sequences are linearly intrapolated in order to obtain a uniform frequency of 100 Hz before being synchronized. From the resulting interpolated and synchronized signal vector $s_i(t)$ first and second order regression coefficients (i. e. first and second temporal derivatives $s_i'(t)$ and $s_i''(t)$) are calculated for every time step t and each component of the low-level signal vector $s_i(t)$. Thus, together with $s_i'(t)$ and $s_i''(t)$, we have $3 \times 6 = 18$ low-level data sequences at this stage.

As mentioned before, an alternative to directly using the 100 Hz low-level signals $s(t) = [s_i(t), s_i'(t), s_i''(t)]$ as inputs for LSTM-based driver state prediction every 10 ms is to compute a set of statistical functionals over longer time windows and use those functionals $x(f)$ as a basis for prediction. Thereby f refers to the index of the frame which contains functionals extracted from time windows of three seconds. As frame rate we use 500 ms resulting in a frame overlap of 2.5 seconds. Depending on whether or not this kind of frame-wise processing is used, either $x(f)$ or $s(t)$ is normalized to have zero mean and variance one. Thereby means and variances are determined from the training set.

The normalized signals $x_n(f)$ or $s_n(t)$ are then used as inputs for the LSTM network, meaning that the individual components of the vectors $x_n(f)$ / $s_n(t)$ represent the activations of the input nodes of the network at a given time step t or frame f. Consequently the LSTM network has as many input nodes as there are components in the vectors $x_n(f)$ and $s_n(t)$, respectively. The number of output nodes of the network corresponds to the number of distinct classes in the prediction task. As detailed in Section VI, we investigate three different classification tasks: the discrimination between two, three, and six different levels of distraction. Thus, our LSTM network has either two, three, or six output nodes. The activation of the output nodes $o(f)$ / $o(t)$ corresponds to the likelihood that the respective class (or distraction level) is observed at a given time step. Note that since the network is trained on discrete class targets rather than on continuous scales for the level of distraction, we do not follow a regression approach, i. e. we do not apply networks with just one output node whose activation indicates the level of distraction. Instead we use a *softmax* output layer (see [26]), enabling the interpretation of the activations of multiple output nodes as probability distribution over the classes. Consequently the output activations sum up to one at each time step. To obtain a prediction $p(f)$ or $p(t)$ of the level of driver distraction at each frame or time step, we simply take the class corresponding to the maximum network output activation.

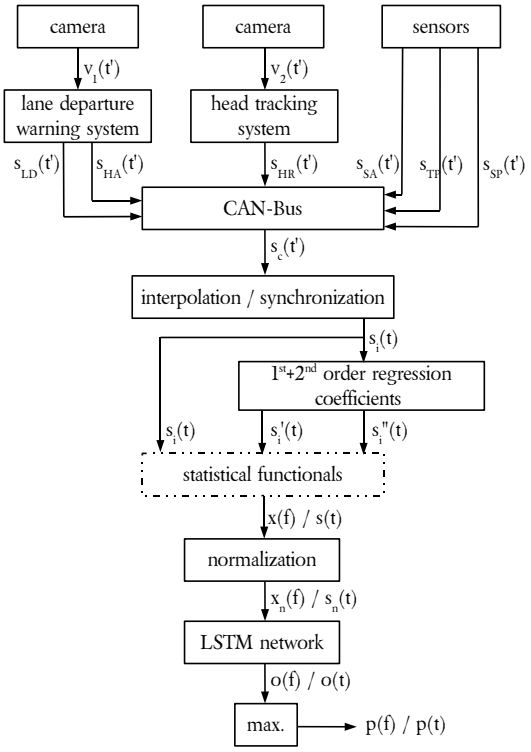

Fig. 2. System architecture of the driver distraction detection system

IV. Feature Extraction

This section will provide insights into the selection of statistical functionals that are computed from the low-level signal vector $s(t)$ in order to obtain a frame-wise feature vector $x(f)$.

As mentioned in Section III, we examine two different strategies for driver distraction detection: firstly, the low-level signals, together with their first and second temporal derivatives (i. e. first and second order delta regression coefficients), are used for *sample-wise* classification every 10 ms. Secondly, *frame-wise* classification is applied by computing statistical functionals every 500 ms from both, the low-level signals and their derivatives (55 functionals per input signal, see Tables I and III) with one frame spanning three seconds. Temporal derivatives of the low-level signals were calculated according to the following formula:

$$s_i'(t) = \frac{\sum_{d=1}^{D} d \cdot (s_i(t+d) - s_i(t-d))}{2 \cdot \sum_{d=1}^{D} d^2} \quad (1)$$

The parameter D was set to one. For the calculation of the second derivative $s_i''(t)$ we simply applied Equation 1 to $s_i'(t)$.

Applying our openEAR toolkit [27], we computed a set of 55 statistical functionals for each of the 18 low-level signals as a basis for the frame-wise classification task. Thus, we obtain a 990-dimensional feature vector for each 500 ms frame.

Using the validation partitions (see Section VI), each, a correlation-based feature subset selection (CFS) was applied to

functionals	abbreviation
Extremes	
maximum, minimum	max, min
range (max - min)	range
distance between maximum and mean	distmax
distance between minimum and mean	distmin
Regression	
linear regression coefficients 1 and 2	lregc1/2
arithmetic mean of linear regression error	mlrege
quadratic mean of linear regression error	qmlrege
quadratic regression coefficients 1, 2, and 3	qregc1/2/3
arithmetic mean of quadratic regression error	mqrege
quadratic mean of quadratic regression error	qmqrege
Means	
arithmetic mean	mean
arithmetic mean of non-zero values	nzmean
arithmetic mean of absolute non-zero values	nzmeanabs
geometric mean of non-zero values	nzgmean
Percentiles	
quartiles 1, 2, and 3 (25 %, 50 %, and 75 %)	q1, q2, q3
interquartile range 1-2, 2-3, and 1-3	iqr1-2/2-3/1-3
Peaks	
mean of peaks	pkmean
distance between mean of peaks and mean	pkmmd
others	
number of non-zero values (normalized)	nnz
zero crossing rate	zcr
mean crossing rate	mcr

TABLE I
LIST OF STATISTICAL FUNCTIONALS COMPUTED FROM EACH LOW-LEVEL SIGNAL VIA THE OPENEAR TOOLKIT [27] TOGETHER WITH THEIR GROUPING INTO CATEGORIES AND THEIR CORRESPONDING ABBREVIATIONS USED IN TABLE II. NOTE THAT THIS TABLE CONTAINS ONLY THOSE FUNCTIONALS WHICH WERE SELECTED AT LEAST ONCE VIA CFS (29 OUT OF 55).

2 classes		3 classes		6 classes	
feature	**#**	**feature**	**#**	**feature**	**#**
HR-min	30	HR-min	30	HR-min	30
HR-pkmmd	30	HR-pkmmd	30	SA-max	30
HR-q1	30	HR-q1	30	HR-q1	30
HR-iqr1-2	30	HR-iqr1-2	30	HR-iqr1-2	30
HR-iqr2-3	30	HR-iqr2-3	30	HR-iqr2-3	30
HR-iqr1-3	30	HR-iqr1-3	30	$\delta\delta$SA-max	30
HR-lregc2	30	HR-lregc2	30	$\delta\delta$SA-min	30
HR-qregc3	30	HR-qregc3	30	HR-mqrege	30
HR-mqrege	30	HR-mqrege	30	SA-min	29
$\delta\delta$SA-nzgmean	30	$\delta\delta$SA-nzgmean	30	$\delta\delta$SA-nzgmean	29
LD-max	28	LD-max	30	HR-iqr1-3	29
HR-q2	27	HR-mlrege	30	HR-lregc2	29
HR-mlrege	26	HR-q2	29	SP-pkmean	29
$\delta\delta$SA-distmax	26	$\delta\delta$SA-min	29	HR-q2	28
HR-mcr	23	$\delta\delta$SA-pkmean	29	HR-mlrege	28
$\delta\delta$SA-pkmmd	23	$\delta\delta$SA-pkmmd	29	HR-qregc3	28
HR-pkmean	22	SA-min	29	$\delta\delta$SA-pkmmd	27
δHR-nzgmean	22	HR-mcr	28	SA-pkmean	26
δHA-pkmean	20	HR-qmqrege	28	HR-mcr	24
HR-qmqrege	19	δHR-nzgmean	28	δHR-nzgmean	24
$\delta\delta$SA-distmin	19	HR-nzmean	25	$\delta\delta$LD-min	24
HR-nzmean	18	SA-max	24	LD-max	23
HR-distmin	17	SP-pkmean	24	δLD-min	23
HA-nzmeanabs	16	SA-pkmean	23	HR-qmqrege	22
HR-qmlrege	16	HR-pkmean	23	δSA-min	22
$\delta\delta$SA-pkmean	16	HR-distmin	22	δSA-max	20
$\delta\delta$SA-range	14	HR-mean	21	δLD-max	19
HR-mean	13	HR-qmlrege	21	$\delta\delta$SA-range	19
$\delta\delta$SA-zcr	13	$\delta\delta$SA-max	21	HR-mean	18
SA-max	12	$\delta\delta$SA-nnz	21	HR-nzmean	18

TABLE II
RANKING OF THE 30 MOST FREQUENTLY SELECTED SIGNAL-FUNCTIONAL COMBINATIONS FOR THE DISCRIMINATION OF TWO, THREE, AND SIX LEVELS OF DISTRACTION. δ AND $\delta\delta$ INDICATE FIRST AND SECOND TEMPORAL DERIVATIVES, RESPECTIVELY. ABBREVIATIONS IN CAPITAL LETTERS INDICATE THE UNDERLYING LOW-LEVEL SIGNAL: STEERING WHEEL ANGLE (SA), THROTTLE POSITION (TP), SPEED (SP), HEADING ANGLE (HA), LATERAL DEVIATION (LD), OR HEAD ROTATION (HR). ABBREVIATIONS IN LOWER CASE LETTERS REPRESENT THE STATISTICAL FUNCTIONALS (SEE TABLE I). NUMBERS DISPLAY THE NUMBER OF FOLDS IN WHICH THE CORRESPONDING FEATURE WAS SELECTED VIA CFS.

these functionals in order to reduce the dimensionality feature space by focussing on the most relevant features [28], [29]. The main idea of CFS is that useful feature subsets should contain features that are highly correlated with the target class while being uncorrelated with each other. The core of CFS is an evaluation function

$$M_S = \frac{k \cdot r_{cf}}{\sqrt{k + k(k-1)r_{ff}}}, \tag{2}$$

where M_S is the rating of a subset S with k features. r_{cf} denotes the mean feature-class correlation and r_{ff} is the average feature-feature inter-correlation. Good subsets of features have highly predictive properties, yielding a high value in the numerator of Equation 2, and a low degree of redundancy among the features, yielding a small value in the denominator. For correlation measurement, the symmetrical uncertainty coefficient is used (as described in [28]). To avoid an exhaustive search in the feature space a greedy hill climbing forward search is applied [29]. In this heuristic search algorithm, each feature is tentatively added to the feature subset, whereas the resulting set of features is evaluated using Equation 2. Once the (so far) best feature set has been chosen, the procedure is repeated. Note that we willfully decided for a filter-based feature selection method, since a wrapper-based technique would have biased the resulting feature set with respect to compatibility to a specific classifier. As termination criterion we considered a maximum of five non-improving nodes before terminating the greedy hill climbing forward search.

Since we arranged our driver distraction estimation experiments in a 30-fold cyclic leave-one-driver-out cross-validation, we conducted the feature selection 30 times for each prediction task (two- three- and six-class problem). On average, 33.8 features were selected for a given classification task and fold (see Table III). Insights into the usefulness of the computed signal-functional combinations can be gained by ranking the features according to the number of folds in which they were selected via CFS. Such a ranking can be found in Table II where the 30 most frequently selected features are listed for each classification task. As assumed, functionals computed from the head rotation signal provide the most reliable features for the detection of driver distraction caused by the operation of the Multimedia Interface. According to Table II, several different functionals such as minimum, mean, distance between the mean of the peaks and the mean, quartiles, interquartile ranges, or linear and quadratic regression coefficients are suited to extract useful information from the head rotation signal. Other frequently selected features are based on the second temporal derivative of the steering wheel angle ($\delta\delta$SA). This indicates

that sudden abrupt movements of the steering wheel – which are necessary to correct the orientation of the car in case the driver does not continuously focus on the street – are a good indicator for distraction. Features computed from the heading angle are mostly selected for the two-class problem and seem less relevant as soon as a finer level of granularity is to be modeled for driver state estimation. By contrast, features based on the lateral deviation signal tend to be rather suited for the six-class task: four out of the 30 most frequently selected features are based on the lateral deviation when modeling six classes, whereas for the two- and three-class task only the maximum lateral deviation (LD-max) is frequently selected. Speed and throttle position are only rarely selected as can also be seen in Table III.

number of funct.		average number of selected features						
type	**total**	**SA**	**TP**	**SP**	**HA**	**LD**	**HR**	**total**
Extremes	3×7	3.4	0.5	0.3	0.5	1.0	1.7	**7.4**
Regression	3×9	0.1	0.1	0.6	0.1	0.2	5.6	**6.7**
Means	3×7	2.3	0.1	0.1	1.2	0.0	2.6	**6.3**
Percentiles	3×6	0.1	0.0	0.3	0.1	0.6	5.0	**6.2**
Peaks	3×4	1.9	0.2	0.4	0.7	0.2	1.7	**5.1**
others	3×22	0.6	0.1	0.1	0.1	0.1	1.1	**2.0**
SUM	**3×55**	**8.4**	**1.1**	**1.8**	**2.7**	**2.0**	**17.8**	**33.8**

TABLE III

LEFT-HAND SIDE: FUNCTIONAL CATEGORIES AND NUMBER OF CALCULATED FUNCTIONALS PER DATA STREAM (EACH STREAM CONSISTS OF THE LOW-LEVEL SIGNAL, FIRST, AND SECOND ORDER REGRESSION COEFFICIENTS); RIGHT-HAND SIDE: AVERAGE NUMBER OF FEATURES SELECTED VIA CORRELATION-BASED FEATURE SELECTION FOR THE INDIVIDUAL DATA STREAMS: STEERING WHEEL ANGLE (SA), THROTTLE POSITION (TP), SPEED (SP), HEADING ANGLE (HA), LATERAL DEVIATION (LD), AND HEAD ROTATION (HR). ALL NUMBERS ARE AVERAGED OVER ALL 30 LEAVE-ONE-SUBJECT-OUT FOLDS AND ALL CLASSIFICATION TASKS.

V. LONG SHORT-TERM MEMORY

This section explains the principle of the Long Short-Term Memory architecture which we will use for RNN-based classification in Section VI. The principle of LSTM allows us to use the (normalized) low-level signals for dynamic classification as an alternative to computing statistical functionals over time windows of fixed length before assigning classes via static classifiers such as Support Vector Machines. Thus, we obtain an estimation of the driver's state for every time step while modeling the temporal evolution of the input signals. The *amount* of contextual information that is incorporated for predicting the driver's state is thereby learned by the network itself and does not have to be specified beforehand.

However, this would not be possible with conventional RNNs since they cannot access long-range context due to the backpropagated error either inflating or decaying over time (the so-called vanishing gradient problem, see [23]). By contrast, Long Short-Term Memory RNNs [24] overcome this problem and are able to model a self-learned amount of context information.

An LSTM layer is composed of recurrently connected memory blocks, each of which contains one or more memory cells, along with three multiplicative 'gate' units: the input, output, and forget gates. The gates perform functions analogous to read, write, and reset operations. More specifically, the cell input is multiplied by the activation of the input gate, the cell output by that of the output gate, and the previous cell values by the forget gate (see Figure 3). The overall effect is to allow the network to store and retrieve information over long periods of time. For example, as long as the input gate remains closed, the activation of the cell will not be overwritten by new inputs and can therefore be made available to the net much later in the sequence by opening the output gate.

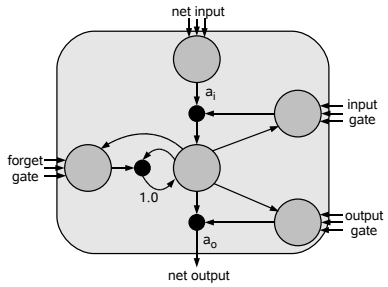

Fig. 3. LSTM memory block consisting of one memory cell: the input, output, and forget gates collect activations from inside and outside the block which control the cell through multiplicative units (depicted as small circles); input, output, and forget gate scale input, output, and internal state respectively; a_i and a_o denote activation functions; the recurrent connection of fixed weight 1.0 maintains the internal state

In our experiments we use *unidirectional* LSTM which exclusively use past context and thus can be applied in a causal on-line detection task. Long Short-Term Memory networks have shown excellent performance in many pattern recognition disciplines [30]–[33].

VI. EXPERIMENTS AND RESULTS

For all experiments a driver independent cross-validation approach was used, whereas the number of folds was equal to the number of participants. In each fold the test set consisted of a single driver (that is, all runs recorded for this person; up to two baselines and eight runs with task) while six other drivers were chosen randomly to form a validation set (containing nine to twelve baselines and 41 to 47 runs with tasks). The data of the remaining persons made up the training set (39 to 42 baselines, 166 to 172 runs with task).

We evaluated three different class distributions, whereas in each of these distributions, the baseline runs are treated as a single class. The runs with distracting tasks either make up another single class (two-class problem) or are split into two or five classes, based upon the individual, subjective rating of the difficulty of the respective task (three-class and six-class problem). In case of the three-class problem, one class consists of all runs rated with difficulties one to three (easy to medium), another one of all runs with difficulties four or five (difficult). In the six-class problem each class corresponds to a single level of difficulty.

In order to investigate the effect of long-range contextual information modeling by using a hidden layer with LSTM architecture (i. e. using memory blocks instead of hidden cells,

LSTM-RNN					
features	**classes**	**accuracy**	**recall**	**precision**	**F1**
low-level sig.	2	91.6 %	89.7 %	90.8 %	90.1 %
low-level sig.	3	54.4 %	62.1 %	63.0 %	62.0 %
low-level sig.	6	43.3 %	39.0 %	38.7 %	38.1 %
functionals	2	96.6 %	95.0 %	97.2 %	**96.0 %**
functionals	3	60.4 %	70.2 %	70.1 %	70.1 %
functionals	6	45.4 %	42.6 %	41.0 %	40.7 %
RNN					
features	**classes**	**accuracy**	**recall**	**precision**	**F1**
low-level sig.	2	74.6 %	60.0 %	68.3 %	63.2 %
low-level sig.	3	42.1 %	46.6 %	46.4 %	45.6 %
low-level sig.	6	37.8 %	30.9 %	30.6 %	29.5 %
functionals	2	94.9 %	92.9 %	95.0 %	**93.8 %**
functionals	3	62.5 %	67.9 %	65.7 %	66.5 %
functionals	6	44.7 %	41.4 %	36.4 %	38.0 %
SVM					
features	**classes**	**accuracy**	**recall**	**precision**	**F1**
functionals	2	91.8 %	88.0 %	90.6 %	**89.1 %**
functionals	3	61.6 %	65.8 %	64.6 %	64.9 %
functionals	6	43.5 %	39.2 %	35.2 %	36.7 %

TABLE IV
CLASSIFICATION OF DRIVER DISTRACTION USING LSTM NETWORKS, STANDARD RNNS, AND SVMS THAT PROCESS EITHER LOW-LEVEL SIGNALS WITH FIRST AND SECOND ORDER REGRESSION COEFFICIENTS OR STATISTICAL FUNCTIONALS OF THE SIGNALS AND REGRESSION COEFFICIENTS: ACCURACY, UNWEIGHTED RECALL, UNWEIGHTED PRECISION, AND (AVERAGE) F1-MEASURE FOR THE SUBJECT-INDEPENDENT DISCRIMINATION OF TWO, THREE, AND SIX LEVELS OF DISTRACTION.

see Section V), we trained and evaluated both, LSTM networks and conventional RNNs using the same configuration. Both, LSTMs and RNNs have an input layer with as many nodes as there are features and a hidden layer with 100 memory blocks or neurons, respectively. Thereby each memory block consists of one cell. The number of output nodes is equal to the number of classes. Each network is trained for up to fifty training iterations, applying an early stopping method. That is, training is instantly terminated if no improvement on the validation set could be achieved within the last ten iterations. To improve generalization, zero mean Gaussian noise with standard deviation 0.4 was added to the inputs during training. The networks were trained with on-line gradient descent, using a learning rate of 10^{-5} and a momentum of 0.9.

For comparison, all experiments employing the computed functionals as input data were repeated using Support Vector Machines with sequential minimum optimization. We applied the LibSVM library, implementing an algorithm that is based on [34]. The best results were achieved with a radial basis function as kernel (gamma kernel coefficient 2^{-6}, cost parameter 1). SVM parameters as well as the choice of the SVM kernel were optimized on the validation data using a grid search and the classification targets corresponding to the two-class task. SVM-based classification of more than two classes was carried out by pairwise coupling according to [35]. Due to past experiences with related classification tasks [32], and due to the discrete classification targets (see Section II), SVM was preferred over regression approaches.

Table IV shows the results for sample-wise classification of driver distraction every 10 ms using the low-level signals together with regression coefficients and for classification every 500 ms applying functionals computed over 3000 ms time windows. Note that due to the imbalance in the class distribution, the F1-measure (harmonic mean of precision and recall) is a more adequate performance measure than accuracy. When using the low-level data, LSTM networks achieve an average F1-measure of 90.1 % for the two-class task and clearly outperform standard RNNs (63.2 %). The major reason for this is the inability of standard RNNs to model long-range time dependencies, which in turn is essential when using the low-level signal as a basis for sample-wise classification. When applying statistical functionals, the temporal evolution of the data streams is captured by the features (to a certain extent), leading to an acceptable performance of RNNs and SVMs (93.8 % and 89.1 %, respectively). Still, the best F1-measure is obtained with LSTM networks (96.0 %). The same holds for the three- and six-class problem, where Long Short-Term Memory modeling leads to an F1-measure of 70.1 % and 40.7 %, respectively, which is remarkable when considering that the participants' ratings of the level of distraction are highly subjective. The performance gap between SVM and LSTM classification can most likely be attributed to the fact that LSTM networks are able to model a flexible and self-learned amount of contextual information which seems to be beneficial for driver state estimation, while the context that is modeled by SVMs is limited to 3000 ms and is exclusively captured by the *features* via statistical functionals and not by the *classifier*.

VII. CONCLUSION

We introduced a technique for on-line driver distraction detection that uses Long Short-Term Memory recurrent neural nets to continuously predict the driver's state based on driving and head tracking data. Our strategy is able to model the long-range temporal evolution of either low-level signals or statistical functionals in order to reliably detect inattention, and can be seen as a basis for adaptive lane-keeping assistance. The amount of contextual information which is used for prediction is thereby learned by the LSTM network itself during the training phase. Experiments revealed that our technique detects inattention with an accuracy of up to 96.6 %, corresponding to an F1-measure of 96.0 %. Thereby we showed that LSTM modeling prevails over conventional RNN networks and Support Vector Machines. From this point of view, an adaption of lane-keeping assistance systems which is based on driver state estimation seems to be a viable and promising approach.

In spite of the high accuracies obtained when operating the proposed driver distraction detection system in defined conditions, such as driving down a relatively straight country road or highway, the output of driver state prediction will of course be less accurate as soon as the driving behavior gets more complex, as for example when changing lanes or turning while driving in a city. Thus, a system for distraction detection as the one presented in this article can only be used if the current driving scenario roughly matches the training data, as it would be the case for most country roads. Similarly, a strong mismatch between the distraction characteristics observed during training and other potential sources of distraction that are

not covered by the evaluation experiments might degrade the system performance and limit the applicability of distraction detection. However, even though negative performance offsets have to be expected under some circumstances and will e. g. justify the additional usage of GPS information as a further indicator of when to activate and deactivate lane-keeping assistance, our experiments show that modeling contextual information is beneficial for driver distraction detection and that the principle of Long Short-Term Memory is an elegant way to cope with this finding.

Future experiments will include the incorporation of *bidirectional* context for incremental refinement of driver state predictions. Bidirectional Long Short-Term Memory (BLSTM) networks can be applied whenever a short latency between observation and estimation is tolerable, since it not only makes use of *past*, but also of *future* context and thus requires a buffer for input data. Bidirectional networks [36] consist of two separate recurrent hidden layers that scan the input sequences in opposite directions and are connected to the same output layer, which therefore has access to context information in both directions. This principle has led to improved accuracies in various sequence labeling tasks [31], [32].

Further, it might be interesting to examine hybrid fusion of the low-level data streams [37] or combinations of RNN-based architectures with Support Vector Machines (e. g. as done in [38]) by classifying activations of RNN output or hidden layers via SVM.

REFERENCES

[1] J. Wang, R. Knipling, and M. Goodman, "The role of driver inattention in crashes; new statistics from the 1995 crashworthiness data system (CDS)," in *40th Annual Proc.: Association for the Advancement of Automotive Medicine*, 1996.

[2] T. Ranney, E. Mazzae, R. Garrott, and M. Goodman, "NHTSA driver distraction research: past, present and future," Washington, DC: National Highway Traffic Safety Administration, Tech. Rep., 2000.

[3] T. Dingus, S. Klauer, V. Neale, A. Petersen, S. Lee, J. Sudweeks, M. Perez, J. Hankey, D. Ramsey, S. Gupta, C. Bucher, Z. Doerzaph, J. Jermeland, and R. Knipling, "The 100-car naturalistic driving study, phase II - results of the 100-car field experiment," Transportation Research Board of the National Academies, Tech. Rep., 2006.

[4] Y. Sugimoto and C. Sauer, "Effectiveness estimation method for advanced driver assistance system and its application to collision mitigation brake system," in *Proc. of 19th International Technical Conference on Enhanced Safety Vehicles*, 2005, pp. 1–8.

[5] R. Freymann, "The role of driver assistance systems in a future traffic scenario," in *Proc. of the 2006 IEEE International Conference on Control Applications*, Munich, Germany, 2006, pp. 2269–2274.

[6] M. Rimini-Döring, T. Altmüller, U. Ladstätter, and M. Rossmeier, "Effects of lane departure warning on drowsy drivers' performance and state in a simulator," in *Proc. of 3. International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design*, Rockport, USA, 2005.

[7] T. Alkim, G. Bootsma, and S. Hoogendoorn, "Field operational test 'The assisted driver'," in *Proc. of Intelligent Vehicles Symposium*, Instanbul, Turkey, 2007.

[8] K. Kozak, J. Pohl, W. Birk, J. Greenberg, B. Artz, M. Blommer, L. Cathey, and R. Curry, "Evaluation of lane departure warnings for drowsy drivers," in *Proc. of Human Factors and Ergonomics Society 50th Annual Meeting*, San Francisco, USA, 2006.

[9] C. Blaschke, F. Breyer, B. Färber, J. Freyer, and R. Limbacher, "Driver distraction based lane-keeping assistance," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 12, no. 4, pp. 288–299, 2009.

[10] K. Torkkola, N. Massey, and C. Wood, "Detecting driver inattention in the absence of driver monitoring sensors," in *Proc. of International Conference on Machine Learning and Applications*, Louisville, USA, 2004.

[11] D. de Waard, K. A. Brookhuis, and N. Hernandez-Gress, "The feasibility of detecting phone-use related driver distraction," *International Journal of Vehicle Design*, vol. 26, no. 1, pp. 85–95, 2001.

[12] 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," *Human Factors*, vol. 48, no. 4, pp. 805–821, 2006.

[13] Y. Liang, M. L. Reyes, and J. D. Lee, "Real-time detection of driver cognitive distraction using support vector machines," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 2, pp. 340–350, 2007.

[14] Y. Liang, J. D. Lee, and M. L. Reyes, "Nonintrusive detection of driver cognitive distraction in real time using bayesian networks," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2018/2007, pp. 1–8, 2007.

[15] Q. Ji, Z. Zhu, and P. Lan, "Real-time nonintrusive monitoring and prediction of driver fatigue," *IEEE Transactions on Vehicle Technology*, vol. 53, no. 4, pp. 1052–1068, 2004.

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

[17] Q. Ji, P. Lan, and C. Looney, "A probabilistic framework for modeling and real-time monitoring human fatigue," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 36, no. 5, pp. 862–875, 2006.

[18] Y. Liang and J. D. Lee, *Driver Cognitive Distraction Detection using Eye Movements*. Springer Berlin Heidelberg, 2008, pp. 285–300.

[19] M. H. Kutila, M. Jokela, T. Mäkinen, J. Viitanen, G. Markkula, and T. W. Victor, "Driver cognitive distraction detection: Feature estimation and implementation," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 221, no. 9, pp. 1027–1040, 2007.

[20] T. Kumagai and M. Akamatsu, "Prediction of human driving behavior using dynamic bayesian networks," *IEICE Transactions on Information Systems*, vol. E89D, no. 2, pp. 857–860, 2006.

[21] A. Pentland and A. Liu, "Modeling and prediction of human behavior," *Neural Computation*, vol. 11, pp. 229–242, 1999.

[22] L. Qiao, M. Sato, and H. Takeda, "Learning algorithm of environmental recognition in driving vehicle," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 25, no. 6, pp. 917–925, 1995.

[23] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," in *A Field Guide to Dynamical Recurrent Neural Networks*, S. C. Kremer and J. F. Kolen, Eds. IEEE Press, 2001.

[24] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[25] K. Young, M. Regan, and M. Hammer, "Driver distraction: A review of literature," Monash University Accident Research Center, Tech. Rep., 2003.

[26] A. Graves, "Supervised sequence labelling with recurrent neural networks," Ph.D. dissertation, Technische Universität München, 2008.

[27] F. Eyben, M. Wöllmer, and B. Schuller, "openEAR - introducing the Munich open-source emotion and affect recognition toolkit," in *Proc. of ACII*, Amsterdam, The Netherlands, 2009, pp. 576–581.

[28] M. A. Hall, "Correlation-based feature selection for machine learning," Ph.D. dissertation, University of Waikato, 1999.

[29] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*, 2nd ed. San Francisco: Morgan Kaufmann, 2005.

[30] M. Wöllmer, F. Eyben, J. Keshet, A. Graves, B. Schuller, and G. Rigoll, "Robust discriminative keyword spotting for emotionally colored spontaneous speech using bidirectional LSTM networks," in *Proc. of ICASSP*, Taipei, Taiwan, 2009.

[31] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Networks*, vol. 18, no. 5-6, pp. 602–610, 2005.

[32] M. Wöllmer, B. Schuller, F. Eyben, and G. Rigoll, "Combining long short-term memory and dynamic bayesian networks for incremental emotion-sensitive artificial listening," *IEEE Journal of Selected Topics in Signal Processing, Special Issue on Speech Processing for Natural Interaction with Intelligent Environments (to appear)*, 2010.

[33] M. Wöllmer, F. Eyben, A. Graves, B. Schuller, and G. Rigoll, "Bidirectional LSTM networks for context-sensitive keyword detection in a cognitive virtual agent framework," *Cognitive Computation, Special Issue on Non-Linear and Non-Conventional Speech Processing*, 2010.

[34] R. Fan, P. Chen, and C. Lin, "Working set selection using the second order information for training SVM," *Journal of Machine Learning Research*, vol. 6, pp. 1889–1918, 2005.

[35] T. Wu, C. Lin, and R. C. Weng, "Probability estimates for multi-class classification by pairwise coupling," *Journal of Machine Learning Research*, vol. 5, pp. 975–1005, 2004.
[36] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, pp. 2673–2681, 1997.
[37] M. Wöllmer, M. Al-Hames, F. Eyben, B. Schuller, and G. Rigoll, "A multidimensional dynamic time warping algorithm for efficient multimodal fusion of asynchronous data streams," *Neurocomputing*, vol. 73, pp. 366–380, 2009.
[38] Y. Yao, G. L. Marcialis, M. Pontil, P. Frasconi, and F. Roli, "Combining flat and structured representations for fingerprint classification with recursive neural networks and support vector machines," *Pattern Recognition*, vol. 36, no. 2, pp. 397–406, 2003.