Online driver distraction detection using long short-term memory

Martin Wollmer, Christoph Blaschke, Thomas Schindl, Björn Schuller, Berthold Farber, Stefan Mayer, Benjamin Trefflich

Angaben zur Veröffentlichung / Publication details:

Wollmer, Martin, Christoph Blaschke, Thomas Schindl, Björn Schuller, Berthold Farber, Stefan Mayer, and Benjamin Trefflich. 2011. "Online driver distraction detection using long short-term memory." *IEEE Transactions on Intelligent Transportation Systems* 12 (2): 574–82. https://doi.org/10.1109/TITS.2011.2119483.

Online Driver Distraction Detection Using Long Short-Term Memory

Martin Wöllmer, *Member, IEEE*, Christoph Blaschke, Thomas Schindl, Björn Schuller, *Member, IEEE*, Berthold Färber, Stefan Mayer, and 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 online 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 (SVMs).**

***Index Terms*—Driver assistance systems, driver state estimation, long short-term memory (LSTM), recurrent neural networks (RNNs).**

I. Introduction

DRIVER inattention is a major factor in traffic accidents. The National Highway Traffic Safety Administration estimates that, in 25% of all crashes, some form of inattention is involved [1]. Distraction (aside from 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 and automotive suppliers will need to find a way to deal with this problem.

One method that aims to minimize 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 recently introduced into the market. These systems track the lane markings in front of the vehicle and compute the time it will take for the vehicle to 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 the 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 those systems are annoying for some drivers is easy to explain. That is, lane-keeping assistance aims to prevent 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 reliably recognize a driver's state, 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 perception, monitoring of driver steering and lane-keeping behavior, and recognition of the driver's involvement in a secondary task itself. In recent years, several techniques trying to estimate driver distraction have been published. However, the majority of approaches are developed and evaluated using data that were captured in a driving simulator and not in a real vehicle, where data are much more noisy and complex than they are in a simulator scenario [10]–[14]. A considerable number of studies concentrate on the detection and modeling of fatigue or stress as important causes for inattention (e.g., [15]–[18]). However, as shown in [12], visual distraction also downgrades driving performance.

To detect distraction or inattention while driving, different classification techniques can be found in the literature. The predominant approach is to use static classifiers such as support vector machines (SVMs) [13], [19]. A promising approach can be found in [20], 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. 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

Manuscript received February 9, 2010; revised August 12, 2010; accepted January 30, 2011. Date of publication March 17, 2011; date of current version June 6, 2011. The Associate Editor for this paper was L. Li.

M. Wöllmer, T. Schindl, and B. Schuller are with the Institute of Human-Machine-Communication, Technische Universität München, 80333 München, Germany (e-mail: woellmer@tum.de; schuller@tum.de).

C. Blaschke, S. Mayer, and B. Trefflich are with the Audi Electronics Venture GmbH, 85080 Gaimersheim, Germany.

B. Färber is with the Human Factors Institute, Universität der Bundeswehr, 85577 München, Germany.

of about 80%. Similar approaches toward driver behavior or driver state estimation that model contextual information via DBNs or Markov models can also be found in [21] and [22]. Other popular classification strategies include the application of fuzzy logic [23], 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 (RNNs) can, in principle, map from the entire *history* of previous inputs to each output. However, 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 [24]). This led to the introduction of long short-term memory (LSTM) RNNs [25]. 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, which is a property that is highly beneficial for predicting the state of a driver.

In this paper, 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 LSTM RNNs, exploiting their ability to capture the long-range temporal evolution of data sequences. We investigate both "samplewise" classification based on low-level signals and "framewise" classification using statistical functionals of the signals. We demonstrate that using low-level signals for driver distraction detection is hardly feasible with conventional RNNs, where the amount of accessible context information is limited.

This paper 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 preprocessing and feature extraction we used. Section V briefly reviews the basic principle of LSTM, whereas Section VI shows experimental results. Conclusions are drawn in Section VII.

II. Database and Signals

To collect data that represent 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 km in the last 12 mo. An Audi A6 was used as the experimental car. The car was equipped with the Audi Multimedia System (see Fig. 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 were 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. Thus, 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.

1) Radio: Adjust the radio sound settings. (Choose the submenu "sound," adjust treble and bass to the middle position, and return to the "radio" menu.)
2) CD: Skip to a specific song (search for the song "sail away," and select it; CD already inserted).
3) Phonebook: Search for a name in the phonebook. (Find the name "Werner Blaschke," make a call, and hang up immediately.)
4) Navigation point of interest: Search for a nearby gas station. (Find the nearest "Esso" gas station, and start route guidance.)
5) Phone: Dial a specific phone number. (Manually dial a number consisting of 11 digits.)
6) Navigation: Enter a city in the navigation device. (Manually enter "Burgholzhausen-Center," and start route guidance.)
7) TV: Switch the TV mode to "PAL." (Change TV-norm from North American to European.)
8) 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. In addition, tasks leading to auditory distraction (e. g. talking to a passenger) were not included in our experiments as they are generally considered to be lower risk activities [26].

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 Fig. 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, six steps have to be done to enter a city in the navigation device.

1) Press the hardkey "NAV."
2) Select "Enter Destination" in a list (one row down).
3) Use the speller nine times to enter the city.
4) Press the control button to confirm the city.
5) Select "Downtown" in a list (one row down).
6) Confirm "Start Navigation."

The procedure for the experiment was given 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). 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. (St2070) between Faistenhaar and Aying, Bavaria). The route is straight (apart from two slight turns), consists of one lane per direction, and leads through a forest. 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 attentively driving 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 s altogether, whereas 9 145.8 s of "distracted" driving were logged. Thus, the average duration of attentive and distracted runs was 59.2 and 41.6 s, respectively. At an average speed of roughly 100 km/h, this corresponds to distances of 1.64 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 of 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 with 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 also model the *degree* of distraction as a six-class problem ("attentive" plus five levels of distraction; see Section VI). For the three-class problem, difficulties 1–3 and 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 an estimate of the subjective 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 to model the driver state on a long-term basis, which, in turn, is desired when using driver state estimations for adaptive lane-keeping assistance.

Six signals were chosen for a first analysis.

1) steering wheel angle (SA);
2) throttle position (TP);
3) speed (SP);
4) heading angle (HA, angle between the longitudinal axis of the vehicle and the tangent on the center line of the street);
5) lateral deviation (LD, deviation of the center of the car from the middle of the traffic lane);
6) 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 [26].) The car's heading angle and its lateral deviation in the lane rely on the amount of attention that 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 [27]. 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 Fig. 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 Fig. 2. Thereby, statistical functionals can be parameters such as extremes, percentiles, and means (see Section IV).

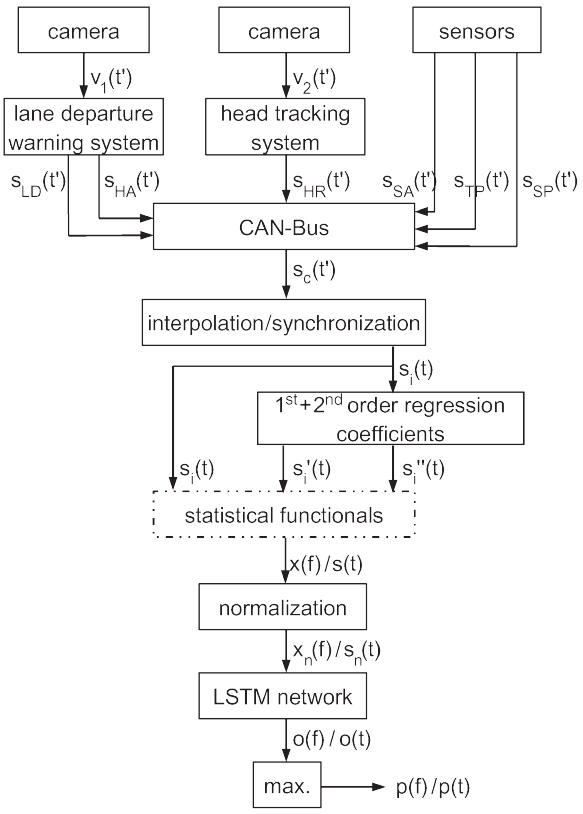

Fig. 2. System architecture of the driver distraction-detection system.

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 $s_{\mathrm{LD}}(t')$ and heading angle $s_{\mathrm{HA}}(t')$. The head rotation $s_{\mathrm{HR}}(t')$ 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 $s_{\mathrm{SA}}(t')$, throttle position $s_{\mathrm{TP}}(t')$, and speed $s_{\mathrm{SP}}(t')$ are captured by the corresponding sensors and sent to the CAN-Bus, together with $s_{\mathrm{LD}}(t')$, $s_{\mathrm{HA}}(t')$, and $s_{\mathrm{HR}}(t')$.

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 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 low-level signals $s(t) = [s_i(t), s'_i(t), s''_i(t)]$ as inputs for LSTM-based driver state classification 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 classification. Thereby, f refers to the index of the frame that contains functionals extracted from time windows of 3 s. As frame rate, we use 500 ms, resulting in a frame overlap of 2.5 s. Depending on whether this kind of framewise 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 classification task. As detailed in Section VI, we investigate three different classification tasks, i.e., the discrimination between two, three, and six different levels of distraction. Thus, our LSTM network has 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 [28]), 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 an estimate $p(f)$ 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.

IV. Feature Extraction

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

As mentioned in Section III, we examine two different strategies for driver distraction detection: First, the low-level signals, together with their first and second temporal derivatives (i.e., first- and second-order delta regression coefficients), are used for *samplewise* classification every 10 ms. Second, *framewise* 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 3 s. Temporal derivatives of the low-level signals were calculated according to the following formula:

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

Parameter D was set to one. For the calculation of the second derivative $s''_i(t)$, we simply applied (1) to $s'_i(t)$.

Applying our openEAR toolkit [29], we computed a set of 55 statistical functionals for each of the 18 low-level signals as a basis for the framewise classification task. Thus, we obtain a 990-dimensional feature vector for each 500-ms frame.

Using the validation partitions (see Section VI), a correlation-based feature subset selection (CFS) was applied to these functionals to reduce the dimensionality of the feature space by focusing on the most relevant features [30], [31]. The main idea of CFS is that useful feature subsets should contain

TABLE I
STATISTICAL FUNCTIONALS GROUPED INTO CATEGORIES WITH ABBREVIATIONS AS USED IN TABLE II

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

features that are highly correlated with the target class while being uncorrelated with each other. The core of CFS is an evaluation function

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

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 intercorrelation. Good subsets of features have highly predictive properties, yielding a high value in the numerator of (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 [30]). To avoid an exhaustive search in the feature space a greedy hill climbing forward search is applied [31]. In this heuristic search algorithm, each feature is tentatively added to the feature subset, whereas the resulting set of features is evaluated using (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 nonimproving 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 classification 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

TABLE II
RANKING OF THE 30 MOST FREQUENTLY SELECTED SIGNAL-FUNCTIONAL COMBINATIONS FOR THE DISCRIMINATION OF TWO, THREE, AND SIX LEVELS OF DISTRACTION. NUMBERS DISPLAY THE NUMBER OF FOLDS IN WHICH THE CORRESPONDING FEATURE WAS SELECTED VIA CFS. 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 FUNCTIONALS (SEE TABLE I)

Two classes		Three classes		Six 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

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 of 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

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 CFS 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

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**

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 tasks, 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.

V. LSTM

This section explains the principle of the LSTM 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 SVMs. 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 back-propagated error either inflating or decaying over time (the so-called vanishing gradient problem; see [24]). By contrast, LSTM RNNs [25] 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 Fig. 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.

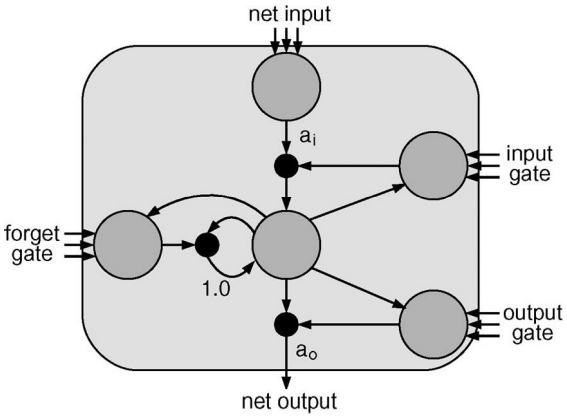

Fig. 3. LSTM memory block consisting of one memory cell. The input, output, and forget gates collect activations from inside and outside the block that control the cell through multiplicative units (depicted as small circles). Input, output, and forget gates scale input, output, and internal states, 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 online detection task. LSTM networks have shown excellent performance in many pattern recognition disciplines [32]–[35].

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), whereas six other drivers were randomly chosen to form a validation set (containing nine–12 baselines and 41–47 runs with tasks). The data of the remaining persons made up the training set (39–42 baselines and 166–172 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 on the individual subjective rating of the difficulty of the respective task (three- and six-class problems). In the case of the three-class problem, one class consists of all runs rated with difficulties ranging from one to three (easy to medium), and another one consists of all runs with difficulties of four or five (difficult). In the six-class problem, each class corresponds to a single level of difficulty.

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; 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 50 training iterations, applying

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

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 %

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 a standard deviation of 0.4 was added to the inputs during training. The networks were trained with online 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 SVMs with sequential minimum optimization. We applied the LibSVM library, implementing an algorithm that is based on [36]. The best results were achieved with a radial basis function as kernel (gamma kernel coefficient of 2^{-6} and cost parameter of 1). SVM parameters and 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 [37]. Due to past experiences with related classification tasks [34] and due to the discrete classification targets (see Section II), SVM was preferred over regression approaches.

Table IV shows the results for samplewise 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 samplewise 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 problems, where LSTM 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 have introduced a technique for online driver distraction detection that uses LSTM 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 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. 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 SVMs. 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 estimation will, of course, be less accurate as soon as the driving behavior gets more complex, like, for example, when changing lanes or turning while driving in a city. Thus, a system for distraction detection as that presented in this paper 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 Global Positioning System 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 LSTM is an elegant way to cope with this finding.

Future experiments will include the incorporation of *bidirectional* context for incremental refinement of driver state estimations. Bidirectional LSTM (BLSTM) networks can be applied whenever a short latency between observation and estimation is tolerable since it uses not only of *past* but of *future* context as well, and thus requires a buffer for input data. Bidirectional networks [38] 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 [33], [34].

Furthermore, it might be interesting to examine hybrid fusion of the low-level data streams [39] or combinations of RNN-based architectures with SVMs (e.g., as done in [40]) 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 *Proc. 40th Annu. Assoc. Advancement Automot. Med.*, 1996, pp. 377–392.

[2] T. Ranney, E. Mazzae, R. Garrott, and M. Goodman, "NHTSA driver distraction research: Past, present and future," Nat. Highway Traffic Safety Admin., Washington, DC, 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," Transp. Res. Board Nat. Acad., Washington, DC, 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. 19th Int. Tech. Conf. Enhanced Safety Vehicles*, 2005, pp. 1–8.

[5] R. Freymann, "The role of driver assistance systems in a future traffic scenario," in *Proc. IEEE Int. Conf. Control Appl.*, 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. 3rd Int. Driving Symp. Hum. Factors Driver Assessment, Train. Vehicle Des.*, Rockport, ME, 2005.

[7] T. Alkim, G. Bootsma, and S. Hoogendoorn, "Field operational test 'The assisted driver'," in *Proc. Intell. Vehicles Symp.*, Istanbul, Turkey, 2007, pp. 1198–1203.

[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. Hum. Factors Ergonom. Soc. 50th Annu. Meeting*, San Francisco, CA, 2006, pp. 2400–2404.

[9] 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. 2009.

[10] K. Torkkola, N. Massey, and C. Wood, "Detecting driver inattention in the absence of driver monitoring sensors," in *Proc. Int. Conf. Mach. Learn. Appl.*, Louisville, KY, 2004, pp. 220–226.

[11] 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.

[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," *Hum. 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 Trans. Intell. Transp. Syst.*, vol. 8, no. 2, pp. 340–350, Jun. 2007.

[14] 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.

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

[16] 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.

[17] 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.

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

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

[20] 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.

[21] 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.

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

[23] 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.

[24] 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. Piscataway, NJ: IEEE Press, 2001.

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

[26] K. Young, M. Regan, and M. Hammer, "Driver distraction: A review of literature," Monash Univ. Accident Res. Center, Victoria, Australia, 2003.

[27] P. Smith, M. Shah, and N. da Vitoria Lobo, "Determining driver visual attention with one camera," *IEEE Trans. Intell. Transp. Syst.*, vol. 4, no. 4, pp. 205–218, Dec. 2003.

[28] A. Graves, "Supervised sequence labelling with recurrent neural networks," Ph.D. dissertation, Techn. Univ. München, Munich, Germany, 2008.

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

[30] M. A. Hall, "Correlation-based feature selection for machine learning," Ph.D. dissertation, Univ. Waikato, Hamilton, New Zealand, 1999.

[31] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*., 2nd ed. San Francisco, CA: Morgan Kaufmann, 2005.

[32] 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. ICASSP*, Taipei, Taiwan, 2009, pp. 3949–3952.

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

[34] 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 J. Sel. Topics Signal Process.*, vol. 4, no. 5, pp. 867–881, Oct. 2010.

[35] 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," *Cogn. Comput.*, vol. 2, no. 3, pp. 180–190, Sep. 2010.

[36] R. Fan, P. Chen, and C. Lin, "Working set selection using the second order information for training SVM," *J. Mach. Learn. Res.*, vol. 6, pp. 1889–1918, 2005.

[37] T. Wu, C. Lin, and R. C. Weng, "Probability estimates for multi-class classification by pairwise coupling," *J. Mach. Learn. Res.*, vol. 5, pp. 975–1005, 2004.

[38] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[39] 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," *Neurocomput.*, vol. 73, no. 1–3, pp. 366–380, Dec. 2009.

[40] 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 Recognit.*, vol. 36, no. 2, pp. 397–406, Feb. 2003.

Martin Wöllmer (M'09) received the Diploma in electrical engineering and information technology from Technische Universität München (TUM), München, Germany.

He works as a Researcher, funded by the European Community's Seventh Framework Program Project SEMAINE (FP7/2007-2013), with TUM, where his current research and teaching activity includes the subject areas of pattern recognition and speech processing. His focus lies on multimodal data fusion, context-sensitive machine learning, and automatic recognition of emotionally colored and noisy speech. His publications in various journals and conference proceedings cover novel and robust modeling architectures such as switching linear dynamic models and long short-term memory recurrent neural nets.

Mr. Wöllmer has been a Reviewer for the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, the IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING, and the IEEE TRANSACTIONS ON AFFECTIVE COMPUTING.

Christoph Blaschke received the Dipl.-Psych. degree from Universitaet der Bundeswehr, Munich, Germany.

He was a Researcher with the Human Factors Institute, Universität der Bundeswehr, Munich, Germany, until the beginning of 2010. Since then, he has been working with the lane-keeping assistance team at Audi Electronics Venture GmbH, Gaimersheim, Germany.

Thomas Schindl received the Diploma in electrical engineering and information technology from Technische Universität München (TUM), München, Germany, in 2010. His diploma thesis dealt with the detection of driver distraction using context-sensitive machine learning. During his bachelor thesis, he specialized in the field of affective computing and investigated the expression of emotions via gait patterns.

He is currently with the Institute of Human-Machine-Communication, TUM. His research interests include automatic control engineering, with emphasis on human–machine interaction.

Björn Schuller (M'04) received the Diploma and Ph.D. degrees in electrical engineering and information technology from Technische Universität München (TUM), München, Germany.

He is currently a Lecturer of pattern recognition with the Institute of Human-Machine-Communication, TUM. He authored more than 200 publications in books, journals, and peer-reviewed conference proceedings. His best-known works advance audiovisual processing in the areas of affective computing and multimedia retrieval.

Dr. Schuller serves as a member of the steering committee of the IEEE TRANSACTIONS ON AFFECTIVE COMPUTING; as Guest Editor and Reviewer for several scientific journals, including the IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE, the IEEE TRANSACTIONS ON MULTIMEDIA, and the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS; and as invited speaker, session organizer and chairman, and program committee member of numerous international conferences. Current project steering board activities include SEMAINE, funded by the European Community, and further projects with such companies as BMW, Continental, Daimler, Siemens, Toyota, and VDO. He has been an invited expert with the W3C Emotion and Emotion Markup Language Incubator Groups and an elected member of the HUMAINE Association Executive Committee.

Berthold Färber received the Ph.D. degree in psychology from the University of Regensburg, Regensburg, Germany, in 1980.

Since 1989, he has been a Full Professor for human factors with the Human Factors Institute, Universität der Bundeswehr, Munich, Germany. He was a partner in many national and European research projects, such as PROMETHEUS, DRIVE, MOTIV, and AKTIV, and in the cluster of excellence Cognition for Technical Systems. His research interests include traffic safety, human factors for advanced driver-assistance systems, and robotics.

Stefan Mayer received the Diploma in computer science from the University of Erlangen-Nürnberg, Erlangen, Germany, focusing on pattern recognition and image analysis, and the Ph.D. degree from Humboldt University of Berlin, Berlin, Germany, for his research activities with the German Aerospace Center, Berlin, on object detection in high-resolution image data from airborne remote sensing.

He is currently with Audi Electronics Venture GmbH (AEV), Gaimersheim, Germany, working on electronics advanced development. He has been responsible for camera-based driver monitoring activities at Audi. He represented AEV in the German research initiative AKTIV (2006–2010) on project driver awareness and safety. His focus is on image analysis and computer graphics in the automotive context.

Benjamin Trefflich received the Dr.-Ing. degree from Technische Universitaet Ilmenau, Ilmenau, Germany.

He was working in the field of driver-camera-based driver monitoring activities until the end of 2008. Since then, he has been responsible for project management and coordination with the Electronics Area, Audi Electronics Venture GmbH, Gaimersheim, Germany.