IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 28, 2020, accepted March 22, 2020, date of publication April 9, 2020, date of current version April 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2986810

Machine Learning and End-to-End Deep Learning for Monitoring Driver Distractions From Physiological and Visual Signals

MARTIN GJORESKI[1,2], MATJA Ž GAMS[1,2], (Member, IEEE), MITJA LUŠTREK[1,2], (Member, IEEE), PELIN GENC[3], JENS-U. GARBAS[3], AND TEENA HASSAN[3]

[1]Jožef Stefan Institute, 1000 Ljubljana, Slovenia
[2]Jožef Stefan Postgraduate School, 1000 Ljubljana, Slovenia
[3]Intelligent Systems Group, Fraunhofer Institute for Integrated Circuits IIS, 91058 Erlangen, Germany

Corresponding author: Martin Gjoreski (martin.gjoreski@ijs.si)

This work was supported in part by the Slovenian Research Agency (ARRS) under Grant U2-AG-16/0672, 0287, the Fraunhofer Society via the Young Research Class 2016 program on 'Cognitive Machines' and by Fraunhofer IIS under the Affective Sensing project. The research collaboration between Jozef Stefan Institute and Fraunhofer IIS is funded by a grant received from the German Academic Exchange Service and ARRS in 2019 (Grant 57451027).

ABSTRACT It is only a matter of time until autonomous vehicles become ubiquitous; however, human driving supervision will remain a necessity for decades. To assess the driver's ability to take control over the vehicle in critical scenarios, driver distractions can be monitored using wearable sensors or sensors that are embedded in the vehicle, such as video cameras. The types of driving distractions that can be sensed with various sensors is an open research question that this study attempts to answer. This study compared data from physiological sensors (palm electrodermal activity (pEDA), heart rate and breathing rate) and visual sensors (eye tracking, pupil diameter, nasal EDA (nEDA), emotional activation and facial action units (AUs)) for the detection of four types of distractions. The dataset was collected in a previous driving simulation study. The statistical tests showed that the most informative feature/modality for detecting driver distraction depends on the type of distraction, with emotional activation and AUs being the most promising. The experimental comparison of seven classical machine learning (ML) and seven end-to-end deep learning (DL) methods, which were evaluated on a separate test set of 10 subjects, showed that when classifying windows into distracted or not distracted, the highest F1-score of 79% was realized by the extreme gradient boosting (XGB) classifier using 60-second windows of AUs as input. When classifying complete driving sessions, XGB's F1-score was 94%. The best-performing DL model was a spectro-temporal ResNet, which realized an F1-score of 75% when classifying segments and an F1-score of 87% when classifying complete driving sessions. Finally, this study identified and discussed problems, such as label jitter, scenario overfitting and unsatisfactory generalization performance, that may adversely affect related ML approaches.

INDEX TERMS Machine learning, deep learning, driver distraction, sensors, facial expressions.

I. INTRODUCTION

Every year, 25,000 people lose their lives on EU roads, and a vast majority of these accidents are caused by human errors. These errors can be avoided with advanced safety features. The monitoring of driver distractions is one such feature that can facilitate the realization of EU's long-term objective of moving close to zero fatalities and severe injuries by 2050.[1] The transition from fully human to autonomous driving often contributes to drivers being less focused, e.g., due to drivers having the freedom to execute additional tasks or due to a potential overloading of sensory activity. On the other hand, automation levels 2 and 3 defined in SAE International's standard J3016 [1] require human attention and readiness to

The associate editor coordinating the review of this manuscript and approving it for publication was Wuliang Yin.

[1]https://ec.europa.eu/commission/presscorner/detail/en/IP_19_1793

take over control in difficult situations or when the vehicle requests. Thus, the detection of distracted driving would be especially valuable in future vehicles, at least until complete autonomy is realized. For example,

Affective computing (also called artificial emotional intelligence) is the ability of technical systems to recognize and process human affective states, which can be used to enrich and facilitate human-computer interaction (HCI) [1]. The recognition of the human physical state using sensors is now mature, e.g., every mobile device is now capable of recognizing activity based on acceleration sensors. However, it is rare for a device to be capable of recognizing human mental states, e.g., stress, mental health, cognitive load, and distractions. Thus, the recognition of the human mental state is the new frontier where the most important research is being conducted. It can be used for services that are directly related to the psychological state and for enhanced HCI.

An important application of affective computing is the detection of human driver distractions [3]. Regan *et al.* [4] consider driver distraction as a subcategory of driver inattention, which is defined as ''diversion of attention away from activities critical for safe driving toward a competing activity, which may result in insufficient or no attention to activities critical for safe driving'' (pp. 1776). Hanowski *et al.* [5] present a list of tasks that could lead to diversion of attention. The list includes dialing and texting on the phone, reading, writing and route checking on a map [6].

Since diverted attention of the driver influences driving safety, the ability to sense the driver's mental state is crucial [3]. These data can be gathered using sensors that are worn by the driver or by using sensors that are embedded in the car, such as video cameras. The recorded data could include behavioral cues, such as facial expressions and gestures, or physiological parameters, such as the heart rate, respiration rate and electrodermal activity. By combining this information with driving parameters and contextual information, safety risks could be estimated and timely alerts could be issued to the driver to avoid accidents and save lives.

This paper presents machine-learning (ML)-based methods for detecting driver distraction using multimodal data. The main contributions of this paper are as follows:

- Statistical analysis for the identification of the best features and modalities for detecting each of four types of distraction, namely, cognitive, emotional, sensorimotor and mixed distraction.
- Comparison of classical ML and end-to-end deep learning (DL) models for driver distraction detection, including an analysis with respect to the size of the input window and the type of the input modality: AUs, emotional activation (EMO), heart rate (HR), breathing rate (BR), nasal electrodermal activity (nEDA) or palm EDA (pEDA).
- Identification and discussion of problems such as label jitter, scenario overfitting and generalization performance that may hinder related ML approaches.

The remainder of the paper is organized as follows: Section II presents the related work. Section III describes the data that are used in this work. Section IV presents the proposed ML and DL methods. Section V elaborates the experiments and the experimental results. Section VI discusses the results, and Section VII concludes the study.

II. RELATED WORK

When analyzing systems for detecting driver distraction, one should consider the distraction types, the input signals and the detection methods. Regarding the distraction types, Gomez *et al.* [8] argued that people differ in terms of their reactions to the same distraction during driving. Thus, the relationship among the distractions, the driver reaction and, consequently, traffic accidents is complex. The distractions can occur in visual, manual or cognitive ways [9], [10]. In this study, cognitive, emotional, sensorimotor and mixed distractions are analyzed.

A. INPUT SIGNALS

The input signals can be direct, namely, measured directly from the driver, or indirect, namely, measured from the vehicle. Vehicle acceleration, steering and braking activities are examples of indirect signals of the driver's state [1]. The related work suggests that methods that use indirect input can be informative for detecting driver distractions. Indirect detection methods rely on the vehicle behavior and are often implemented in recently produced cars. Aksjonov *et al.* [24] presented a method for detecting the driver's distraction by monitoring lane maintenance and speed performance on specified road segments. Saito *et al.* [25] proposed an assistance system for prediction the driver's state based on the lane departure duration. Apostoloff and Zelinsky [26] studied the driver's attention to lane maintenance task. Castignani *et al.* [27] developed a system, namely, SenseFleet, that can identify risky driving events by examining the acceleration, braking and steering activities of the driver. Similarly, Pavlidis *et al.* [7] presented a statistical analysis of the relation between driver distractions and the speed, acceleration, brake force, steering and lane position. Wang *et al.* [28] proposed a forward collision warning algorithm that depends on the driver's braking activity. However, if the vehicle is in an autonomous-driving mode, the indirect inputs will not reflect the driver's behavior; instead, they will reflect the behavior of the algorithm for autonomous driving. Additionally, cars may be easily retrofitted with systems that use direct inputs. Thus, this study focuses on input signals that are measured directly from the driver using physiological sensors and visual analysis.

The direct input signals can be divided into two subgroups: (i) visual measurements, such as eye gaze, pupil diameter, head pose, facial expressions and driving posture, (ii) and physiological signals, such as electroencephalogram (EEG), electrooculogram (EOG), electrocardiogram (ECG), electromyogram (EMG), photoplethysmogram (PPG) and electrodermal activity (EDA) signals. The physiological

measurements provide important cues regarding the driver's state, such as his or her drowsiness and stress levels. Lee and Chung [11] evaluated eye tracking and PPG features with a dynamic Bayesian-network-based framework for the detection of driver drowsiness. Lin *et al.* [12] measured the drowsiness of the driver by using EEG signals. They decreased the number of EEG features by using the principal component analysis (PCA) method. Then, these features were fed into a linear regression model for the estimation of the drowsiness level. In addition to the EEG signals, Khushaba *et al.* [13] analyzed the drowsiness of the driver by using EOG and ECG signals. Multiple modalities, such as ECG, EMG, EDA and the respiration rate, have also been used to detect the stress level [14]. In various studies, the physiological sensors were integrated into driving equipment. For example, Singh *et al.* [15] used ECG signals that were measured via electrodes that were placed on the seat and seatbelt. Similarly, Lee *et al.* [16] measured ECG signals via electrodes that were placed on the steering wheel. Additionally, they derived the respiratory rate and HR variability from the ECG signals and used PPG that was measured from the driver's finger.

Visual measurements give the driver more freedom than physiological measurements that are obtained using wearable sensors. Bergasa *et al.* [17] measured the degree of eye closure, the eye closure duration, the blinking and nodding frequencies, and the head pose and conducted eye tracking, and they used these data to estimate the driver's state. Omidyeganeh *et al.* [18] argued that yawning is an important characteristic for estimating driver drowsiness. They used face and mouth features to detect yawning. Vicente *et al.* [19] proposed an eyes-off/on-the-road detection system that is based on head pose and eye gaze estimation. Murphy-Chutorian and Trivedi [20] argued that the driver's head pose is a strong indicator of his or her current focus of attention. Similarly, Smith *et al.* [21] analyzed the driver's attention from head- and face-related features. The hand position was also proposed as an indicator for detecting driver distraction.

In the related studies, there is no consensus regarding the input signals for the detection of driver distraction. Thus, in this study, experiments with both data from physiological sensors and data from video-based sensors were made. The physiological data include pEDA, the HR and the BR. The visual data include nEDA (extracted from data that were captured using a thermal camera), eye tracking data (x-y positions and the pupil diameter), head pose, facial expressions and emotional activation.

B. DISTRACTION DETECTION METHODS

Sikander and Anwar [29] grouped the methods for detecting driver distraction into three subgroups: mathematical models, rule-based models and models that are based on ML algorithms. Most mathematical models are designed for predetermined setups, such as workplace and factory worker workloads. These models consider circadian cycles, sleep history, duration of sleep and wakefulness for the detection of fatigue and performance [30]. For example, the System for Aircrew Fatigue Evaluation (SAFE) is based on such mathematical models [31]. Regarding the rule-based systems, Lee *et al.* [16] derived if-then rules and applied kernel fuzzy-C-mean to detect driving distractions. Azim *et al.* [32] proposed two-layered rule-based systems that were based on eye and mouth state information, where each layer had its own if-then rules.

The most advanced methods for monitoring driving distractions are based on ML algorithms. These methods can be classical, deep or a combination of both classical and DL [29]. Goel *et al.* [10] evaluated random forest, Naïve Bayes, SVM and decision tree for the detection of driving distraction. Random forest outperformed all the other strategies. Lee *et al.* [23] analyzed hand movements that were detected by acceleration sensors in a smartwatch. They calculated features and fed them into a support vector machine (SVM) classifier.

In addition to the classical feature-based ML methods, end-to-end DL methods, namely, methods for which feature extraction is not required and raw inputs are fed into the models, were also proposed. Masood *et al.* [6] detected distractions and causes of the distractions by using CNNs. Majdi *et al.* [34] developed Drive-Net, which combines CNN and Random Forest for the detection of the distraction categories in images. Yan *et al.* [22] used CNNs to detect various driving postures in images, such as normal driving, cell phone call, eating and smoking. Hssayeni *et al.* [35] compared two approaches for distracted driving detection: the use of traditional handcrafted image-based features along with SVM and the use of features from three end-to-end CNNs, namely, AlexNet, VGG-16 and ResNet-152. ResNet and VGG-16 outperformed AlexNet by almost 10%. The feature-based SVM realized much lower accuracy than the CNNs. Similarly, Koesdwiady *et al.* [33] used VGG-19.

All the end-to-end DL approaches use image data as input and are based on available DL architectures that have been successfully applied on images (e.g., AlexNet, VGG-16 and ResNet-152), and most focus on only one architecture. The DL architectures in this study use 1D signals as inputs; thus, specialized DL architectures for multimodal time-series data were investigated. As few studies have been conducted on end-to-end learning on 1D signals, seven DL architectures were compared in this study. To the best of our knowledge, this is the first study on the detection of driver distraction that analyzes end-to-end learning on signals using 1D convolutions and long short-term memory neural networks (LSTMs). Additionally, the DL architectures were compared to state-of-the-art classical ML algorithms using an extensive set of features. Comparison among different features/modalities for detecting driver distraction with both the classical and the DL models was also made.

TABLE 1. Layout of the driving sessions. Red represents the driving segments that include an external distraction. N represents normal driving, S represents driving under a distraction and F represents a brake failure event. In the experiments, the brake failure event is regarded as driving under a distraction.

Driving session	Session segment				
Type	1	2	3	4	5
Cognitive drive (CD) −Analytical questions	N	S	N	S	N
Emotional drive (ED) − Emotional questions	N	S	N	S	N
Sensorimotor drive (SD) − Texting on smartphone while driving	N	S	N	S	N
Normal drive (ND) − Same speed limit, heavy traffic and construction blockades	N	N	N	N	N
Relaxed drive (RD) − Same speed limit with light traffic and smooth lane changes.	N	N	N	N	N
Failure-Normal drive (FDN) − Unexpected failure in vehicle leading to a collision course.	N	N	F		
Failure-Loaded drive (FDL) − Unexpected failure in combination with mixed distractions	N	S	F		

TABLE 2. 46 channels of information that were used in the study, grouped per modality.

Modality	Channels
Physiological sensors (3)	pEDA, HR and BR
Thermal camera (1)	nEDA
Eye tracking (4)	eye position X and Y coordinates, left and right pupil diameters
Video camera (8)	emotional response: anger, contempt, disgust, fear, joy, sad, surprise, neutral
AUReader (30)	AU01InnerBrowRaiser, AU02OuterBrowRaiser, AU04BrowLowerer, AU05UpperLidRaiser, AU06CheekRaiser, AU07LidTightener, AU09NoseWrinkler, AU10UpperLipRaiser, AU11NasolabialDeepener, AU12LipCornerPuller, AU13SharpLipPuller, AU14_Dimpler, AU15LipCornerDepressor, AU16LowerLipDepressor, AU17ChinRaiser, AU20_LipStretcher, AU23LipTightener,AU24_LipPresser, AU25LipsPart, AU26_JawDrop, AU27_MouthStretch, AU43EyesClosed, PositiveValence, NegativeValence, RotationX, RotationY, RotationZ, TranslationX, TranslationY, TranslationZ

III. DATA DESCRIPTION

The experimental data are obtained from a study by Pavlidis *et al.* [7]. In the study, they analyzed the driving behaviors of 68 volunteers in a driving simulator under a variety of distractions. Each volunteer had several driving sessions, which included a normal driving session without distractions and sessions under cognitive, emotional, sensorimotor and mixed distractions. The experimental design and the specific stressors are is presented in Table 1.

Pavlidis *et al.* [7] analyzed the relations between the distractions and various driving parameters, such as the speed, acceleration, brake force, steering and lane position. From the physiological response, only nEDA [36], [37] was analyzed. In this study, the overall physiological and affective responses in relation to the external distractions were analyzed. The physiological response includes nEDA, pEDA, HR, BR and eye tracking data. The affective response includes emotions, facial expressions and the head pose.

The physiological response, which was measured using physiological sensors, and the emotional response, which was extracted from facial videos using a software that outputs probability estimates for eight prototypical emotions, were already provided in the dataset. As an addition, the facial expressions in the form of AUs and the head pose were extracted using the facial-expression-analysis software that was presented in Hassan *et al.* [38], which is hereafter referred to as AUReader. AUReader estimates the intensities of 22 facial action units (AUs) using a dynamic state estimation framework that fuses viscoelastic models for facial muscle motion with facial shape and appearance information. AUs are basic facial movements that can be visually distinguished and are defined in the facial action coding system [39], [40]. AUs are produced by a single facial muscle or a group of facial muscles [39], [40], [41]. For example, AU12 represents the action of raising the lip corners (as in a smile) and is produced by the facial muscle 'zygomaticus major'; AU25 represents the mild parting of lips and is produced by either 'depressor labii inferioris' or 'orbicularis oris'; and AU27 represents the stretching or wide opening of the mouth, which is produced by the pterygoids and digastric muscles [39], [40], [41]. Images that show the expressions of AUs are available in [41]. In this study, each facial video in the dataset [7] was analyzed using AUReader to obtain the 3D head pose and AU intensity estimates for each frame in the video.

IV. METHOD

A. PREPROCESSING, FEATURE EXTRACTION and CLASSICAL MACHINE LEARNING

After the extraction of AUs, 46 channels of information (see Table 2) were available: nEDA, pEDA, HR, BR and eye

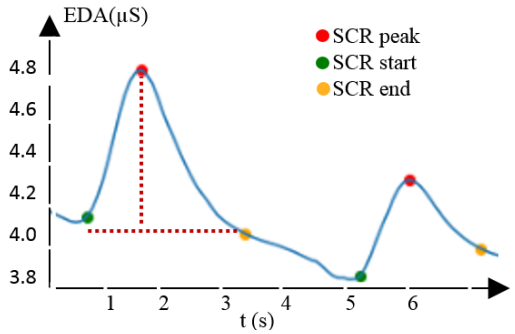

FIGURE 1. Example EDA signal with two skin conductance responses (SCRs). The horizontal (red) dotted line on the first SCR represents the SCR amplitude. The vertical (red) dotted line on the first SCR represents the SCR duration.

tracking data (4 channels), emotional response (8 emotions/channels) and AUReader data (30 channels). First, all channels were resampled with a sampling frequency of 1 Hz. Next, following the normalization procedure that was used by the dataset creators [1], all channels were normalized via an unsupervised person-specific approach. The normalization function was as follows:

$$Sn_{ij} = \log(S_{ij}) - \log(O_{ij})$$

where O_{ij} represents the overall average value for the i^{th} person and the j^{th} channel, S represents the raw data segments, and Sn represents the normalized data segments. The normalized signals of each driving session were segmented into smaller windows.

Experiments were conducted with windows from 20 seconds up to 80 seconds with a stride of 5 seconds. The segmented data were used as the input to the DL models. For the classical ML models, features were extracted from the segmented data and were used as input to the models.

For each window, the following statistical features were extracted from each channel: the mean, standard deviation, skewness, kurtosis, mean of the first derivative, mean of the second derivative, 25^{th} and 75^{th} percentiles, inter-quartile range, difference between the minimum and the maximum values and coefficient of variation.

Additional features were extracted for the pEDA and the nEDA signals using skin conductance response (SCR) analysis (see Figure 1). This type of feature/analysis is proven to be useful for the detection of stressful conditions in driving scenarios [14] and in practice [42]. The SCR features for each window were the power of the EDA signal, the number of SCRs per second, the power of the SCRs, the sum of the signals' components that have positive derivative, the ratio between the positive derivative and the negative derivative, the mean value of the derivative of the tonic component (the slowly changing EDA component), the mean value of the difference between the raw signal and the tonic component, the total spectral power of the signal in five frequency bands between 0 Hz and 0.6 Hz with a 0.1-Hz span, the amplitude increase of the largest SCR (from the SCR start time to the SCR peak), the amplitude decrease of the largest SCR peak, the largest SCR increase time, the largest SCR decrease time, the ratio of the increase time and the decrease time of the largest SCR peak, the largest SCR duration, the largest SCR peak increase and decrease slope, the average amplitude increase and decrease of all SCR peaks, and the average amplitude change of all SCRs.

For the classical models, the ML algorithms were used as implemented in the scikit-learn ML toolkit [43]. For each algorithm, parameter tuning was conducted using the following procedure: First, the parameter settings were randomly sampled from distributions that were predefined by an expert. Next, models were constructed with the specified parameters and evaluated using internal k-fold cross-validation on the training data. The search procedure was repeated 10 times. The averaged results are reported in Section V. Experiments were conducted with the following ML algorithms: decision tree [44], RF [45], naïve Bayes [46], KNN [47], SVM [48], bagging [49], adaptive boosting (AdaBoost) [50] and extreme gradient boosting (XGB), which is an updated boosting algorithm. Decision trees were used as the base model for all the ensemble algorithms.

B. DEEP LEARNING

DL represents a class of ML algorithms that use a cascade of multiple layers of nonlinear processing units, which are typically neurons [51]. The first layer receives the input data, and each successive layer accepts the output from the previous layer as input.

The basic strategy dates back to 1943, when McCulloch and Pitts created the first computational model of neural networks (NNs), which was based on threshold logic [52]. Currently, large processing power and memory storage are relatively affordable, and DL models are used to solve complicated artificial intelligence (AI) tasks (e.g., in computer vision, language, biomedicine, and autonomous driving).

1) FULLY CONNECTED NEURAL NETWORKS

A fully connected (FC) NN is a cascade of multiple layers of nonlinear processing units, where each unit receives input from the previous layer. In a typical FC NN, layer i computes an output vector z_i as follows:

$$z_i = f(b_i + W_i z_{i-1}) \quad (1)$$

where b_i (biases) and W_i (weights) are the parameters for the i^{th} layer, z_{i-1} is the output vector of the previous layer and z_0 is the input data. The activation function f can be a rectified linear unit (ReLU) [53]:

$$f(c) = \max(0, c) \quad (2)$$

or another nonlinear function, such as sigmoid or tanh. For classification problems, the final output layer (z_{F_j}) typically uses a softmax activation function.

$$z_{F_j} = softmax(b_i + W_i z_{i-1}) \quad (3)$$

where j represent the j^{th} row of the weights W_i. The softmax function has the following useful property:

$$\sum_j z_{F_j} = 1 \tag{4}$$

and it is always positive; thus, it can be used as an estimator for the probability that an input pattern x belongs to the j^{th} class for a specified problem:

$$P(Y = j|x) \tag{5}$$

The parameters of the network (b and W) are learned using an optimization algorithm, such as gradient descent [54]. For a binary classification problem, the binary cross-entropy is typically used as a loss function, which is minimized over the pairs of input data/labels (x, y) and predictions p.

$$L = -(ylog\,(p) + (1 - y)\log\,(1 - p)) \tag{6}$$

2) CONVOLUTIONAL NEURAL NETWORKS

CNNs are a type of NNs that are designed with three main architectural strategies to ensure various degrees of shift-, scale- and distortion-invariance. This is realized by utilizing (i) local receptive fields, namely, each unit in a layer receives input from a set of neighboring units in the previous layers; (ii) shared weights, namely, units in a layer are organized in groups and all units in the same group share the same set of weights [57], [58]; and (iii) spatial or temporal sampling, namely, if the input is shifted, the feature map output will also be shifted [55]. In addition, due to the specified architecture (parameter sharing and local connections), the CNNs have far fewer connections and parameters to train, while their theoretical best performance is likely to be only slightly worse than that of FC NNs [56].

3) LONG SHORT-TERM MEMORY

Long short-term memory (LSTM) NNs are a type of recurrent neural networks (RNNs), which are networks with memory mechanisms that enable information to persist through time in the model. LSTMs were introduced by Hochreiter and Schmidhuber [67] in 1997. The main processing unit is an LSTM cell, which contains three main gates that regulate the internal cell state and the cell's output. The first gate decides what information should be forgotten (the forget gate) at time t. The decision is made by a sigmoid function, which is applied over the current input x_t and the previous cell output h_{t-1} (Equation 7). The output of the sigmoid function is a number that is between zero and one, where zero corresponds to no propagation.

$$f_t = \sigma(b_f + W_{fx}x_t + W_{fh}h_{t-1}) \tag{7}$$

Next, the input gate (Equation 8) decides what input information will be passed to the output gate via another sigmoid function. The candidate values $\hat{C}_t$ for the new cell state are calculated by a tanh layer (Equation 9). The output of the tanh layer is always between –1 and 1. The new cell state (C_t) is calculated by multiplying the old state C_{t-1} by f_t to forget some of the previous information and by adding the element-wise product $i_t * \hat{C}_t$, which consists of the candidate values $\hat{C}_t$, scaled by i_t (Equation 10).

$$i_t = \sigma(b_i + W_{ix}x_t + W_{ih}h_{t-1}) \tag{8}$$

$$\hat{C}_t = tanh(b_c + W_{cx}x_t + W_{ch}h_{t-1}) \tag{9}$$

$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t \tag{10}$$

Finally, the output gate decides which parts of the cell state (C_t) it is going to output (propagate) via another sigmoid layer (Equation 11), and the final output of the cell (Equation 12) is calculated by applying tanh on the current cell state and scaling it with o_t from Equation (11).

$$o_t = \sigma(b_o + W_{ox}x_t + W_{oh}h_{t-1}) \tag{11}$$

$$h_t = o_t * tanh(C_t) \tag{12}$$

The equations above represent the main strategy of LSTMs. Additionally, there are many variations of RNNs and LSTMs [68], [69], [70].

4) DEEP LEARING ARCHITECTURES

DL realized a breakthrough performance at solving pattern recognition problems [59], especially in image [56], [60], [63] and natural language processing (NLP) [61], [62]. For example, DL was used to realize image super resolution [64]. In another study, DL was used for ''seeing in darkness'', which is a technique for reconstructing and brightening dark images [65]. For NLP, Google introduced BERT – a state-of-the-art method for ''language understanding'' [66]. However, DL architectures for signal processing have not yet realized such a breakthrough and designing them remains challenging, especially for problems with limited data. The layered structure of the NNs enables the construction a variety of DL architectures by combining layers. For example, ConvLSTM stacks CNN layers on top of LSTM layers, namely, the input is received by the CNN layers and propagated to the LSTM layers. In addition to the vertical stacking, one can also experiment with horizontal stacking. For example, for a 2-channel dataset, one can use a ConvLSTM for each channel and later fuse the outputs of the two ConvLSTMs using an FC layer. Which DL architecture is most suitable depends on the dataset; thus, extensive experimentation is required.

Figure 2 presents the two fusion approaches that are evaluated in the experiments. The early-fusion approach merges all 46 channels at the input regardless of the modality. Then, the merged input data are fed into DL layers. The DL layers can be FC layers, CNN layers or LSTM layers. The mid-fusion approach uses DL layers that are specific for each modality, and later, the modality-specific layers are fused using a general DL. The early-fusion approach learns shared weights for all input modalities, whereas the mid-fusion approach initially learns separate weights for each modality (represented by purple squares in Figure 2) and later learns shared weights (represented by orange squares in Figure 2). An additional DL architecture that is evaluated in the experiments is the spectro-temporal ResNet (STRNet), which is an architecture that was successfully applied on sensor data

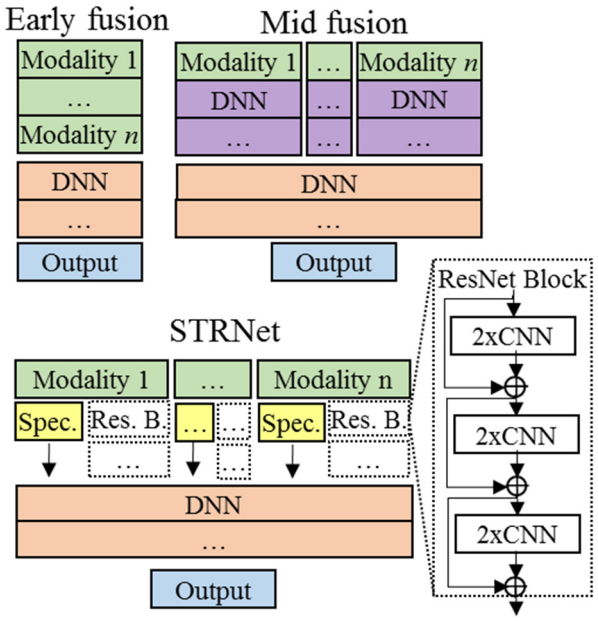

FIGURE 2. Two types of DL fusion approaches that are used in the study: early fusion and mid-fusion. The spectro-temporal ResNet (STRNet) is a special case of the mid-fusion approach.

TABLE 3. Deep learning architectures that are used in the study.

Classifier	Architecture
eCNN	2 x CNN(128) - FC(64)
eLSTM	2 x LSTM(128) - FC(64)
eConvLSTM	2 x CNN(128) - LSTM(128) - FC(64)
mCNN	N x 2 x CNN(128) - FC (64)
mLSTM	N x 2 x LSTM(128) - FC(64)
mConvLSTM	N x 2 x CNN(128) - LSTM(128) - FC(64)
STRNet	N x [3 x (2 x CNN(64))] - FC (64)

in previous study on human activity recognition from smartphone sensors [71], for chronic heart failure detection from heart sounds [72], and for blood pressure estimation from photoplethysmogram (PPG) data [73].

STRNet is a special type of mid-fusion network in which each modality is associated with two branches: one that evaluates the raw sensor signal in the time domain using residual blocks [74] and another that evaluates a spectral representation of the signal. Toward the end of the network, the two branches, namely, the spectral and the temporal branches of each modality, are merged using FC layers.

The structures of the DL architectures that are used in this study are presented in Table 3. There are three early-fusion architectures (eCNN, eLSTM and eConvLSTM) and four mid-fusion architectures (mCNN, mLSTM and mConvLSTM and STRNet). For example, the architecture "2 x CNN(128) - FC(64)" contains two CNN layers, each with 128 filters, and one FC layer with 64 neurons. N represents the number of input channels. All DL architectures contain batch normalization layers [75] to reduce the internal covariance shift, ReLU activation layers [53] to accelerate the training process; maximum pooling layers for dimensionality reduction, and a final softmax layer, which outputs the estimated class probability for distracted vs. not distracted driving. The DL architectures are available online https://repo.ijs.si/martingjoreski/driving-distractions/tree/master/DL%20architectures. All DL models were trained by minimizing the binary cross-entropy loss function using the Adam optimizer with a learning rate of 10^{-5} and a decay of 10^{-3}. The batch size was set to 256 with a maximum number of training epochs of 30.

V. EXPERIMENTS

First, a statistical analysis of the input signals was conducted to analyze the relations between the modalities and the driving distractions. Next, ML analysis was conducted to compare classical ML and DL for the detection of driving distractions. Next, ML analysis was conducted to compare classical ML and DL for the detection of driving distractions.

For the ML analysis, the data of the first 10 subjects were used as the test set (close to 20% of the overall data), and the data of the remaining subjects were used as the training set. Thus, the classifiers are subject-independent. Each ML algorithm was evaluated in the construction of two types of classifiers:

- A window classifier: Outputs a prediction whether distraction was detected for each input window (binary classification). This classifier would be useful for monitoring driver distractions in real time;
- A session classifier: Outputs only one prediction per driving session, namely, each driving session is classified as 'with distractions' or 'without distractions'. The decision is based on the predictions of the window classifier that are obtained using a threshold logic. The thresholds were optimized for each classifier using cross-validation on the training set. This classifier would be useful for the offline determination of whether there was a distraction present during the past driving session.

The ground-truth labels were determined using the following rules: (i) the instances for the window classifiers are labeled as positive, namely, distracted driving should be detected, if a distraction was present in at least 5 seconds of the input window and (ii) the instances for the session classifiers are labeled as positive if a distraction was present for at least 5 seconds of the overall driving session. For the window classifiers, one instance is one window (segment) that was extracted using an overlapping sliding window with a 5-second stride; thus, a prediction is output every 5 seconds.

For the session classifiers, one instance is one session. For example, Table 4 summarizes the experimental data (instances) that are produced after using an overlapping sliding window of 60 seconds with a stride of 5 seconds.

Experiments were conducted with window sizes from 20 to 80 seconds. F1-score was used to evaluate the classifiers

TABLE 4. Experimental data. The sizes of the training and test subsets for the window classifiers and for the session classifiers.

Instances	Subset	Distraction		Majority
		No	Yes	
Window	Test	3724	2135	64 (%)
	Training	13144	8023	62 (%)
Session	Test	21	34	62 (%)
	Training	66	128	66 (%)

(Equations 13 to 15):

$$Precision = 100 * \frac{Correctly\ detected\ distrsction\ instances}{Overall\ predicted\ distractions} \quad (13)$$

$$Recall = 100 * \frac{Correctly\ detected\ distrsction\ instances}{Overall\ distraction\ instances} \quad (14)$$

$$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (15)$$

A. INPUT ANALYSIS

In the initial dataset study [1], the authors showed that there is a statistically significant difference in the mean values of the nEDA when measured in the normal segments of the driving sessions, compared to the distracted segments of the same driving sessions. Inspired by that analysis, statistical tests were conducted in this study to determine whether such a statistically significant difference is present for the remaining features in the experiments. For the statistical analysis, the Wilcoxon signed-rank test was used, which is an alternative to the paired Student's t-test that lacks the t-test's normality assumption on the distribution of the paired differences. The Wilcoxon test is a non-parametric statistical hypothesis test that is used to determine whether two paired samples are sampled from the same distribution [76]. In this experimental setting, one sample contains values of a specified feature that was extracted from the normal segments of the driving sessions, and the other sample contains values for the same feature that were extracted from the distraction segments of the same driving session. Informative features should differ in terms of their distributions when conditioned on the type of the segment (with vs. without distraction). The tests showed for 177 of the 562 features, the test p-value was smaller than 0.001; these are named "informative features". Table 5 presents the top three modalities for each type of driving session (ED, SD, CD, FDL and FDN) and for all driving sessions (Overall). The modalities are ranked using the ratio of informative features per modality. According to the table, nEDA is ranked among the top 3 for each driving session. The emotions are ranked among the top 3 for five out of the six types of driving sessions. The facial action units (AUs) are ranked among the top 3 for three types of driving sessions (ED, CD and FDN). For recognizing the mixed distractions in the failure sessions (FDL and FDN), nEDA is the most informative modality. Overall, when all driving sessions are joined and the statistical tests are conducted for normal segments vs. distraction segments, the two most informative modalities are the recognized emotions and nEDA. This is followed by the facial AUs in the third position.

TABLE 5. Top three modalities for each type of driving session and overall) ranked according to the ratio of informative features.

Session	Modality	Informative features (%)
ED	Emotion	22/87 (25.3)
	nEDA	9/39 (23.1)
	Face AUs	27/305 (1.8)
SD	Breathing	4/11 (36.4)
	Eye Tr.	8/32 (25.0)
	nEDA	6/39 (15.4)
CD	Emotion	22/87 (25.3)
	nEDA	5/39 (12.8)
	Face AUs	29/305 (9.5)
FDL	nEDA	4/39 (10.3)
	Breathing	1/11 (9.1)
	Emotion	2/87 (2.3)
FDN	nEDA	3/39 (7.7)
	Emotion	2/87 (5.1)
	Face AUs	3/305 (1.0)
Overall	Emotion	47/87 (54)
	nEDA	18/39 (46.2)
	Face AUs	47/305 (15.4)

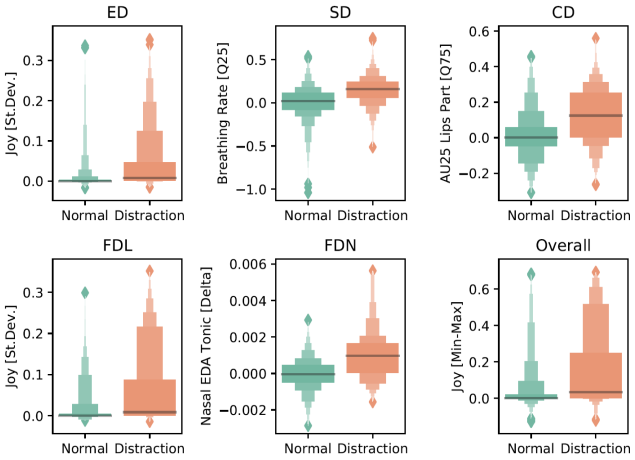

FIGURE 3. Distributions of the most informative features, namely, the features with the smallest p-value, for each type of driving session (ED, SD, CD, FDL, FDN) and overall.

Figure 3 presents the distributions of the most informative features, namely, the features with the smallest p-value for each type of driving session (ED, SD, CD, FDL, FDN) and for all driving sessions (Overall).

The distributions are represented as boxenplots (letter-value-plots), which provide a better representation of the distribution of the data than boxplots when outlier values are present [77]. According to the figure, for recognizing an emotional distraction (ED), the most informative feature is the standard deviation of the activation of the emotion "joy".

TABLE 6. Evaluation results for the classical ML classifiers and DL models. F1-scoress of the window classifiers (F1) and F1-score of the session classifiers (F1-s).

Type	Classifier	F1 (%)	F1-s (%)
classical ML	RF	67	84
classical ML	**GB**	**73**	**87**
classical ML	KNN	50	78
classical ML	NB	63	74
classical ML	DT	56	66
classical ML	Bagging	64	82
classical ML	**XGB**	**72**	**88**
end-to-end DL	eCNN	65	68
end-to-end DL	**eLSTM**	**67**	**75**
end-to-end DL	eConvLSTM	65	69
end-to-end DL	mCNN	64	69
end-to-end DL	mLSTM	64	70
end-to-end DL	mConvLSTM	62	69
end-to-end DL	**STRNet**	**67**	**80**

Thus, during the distraction segments, the subjects showed an increased standard deviation of this emotion. Second, for recognizing a sensorimotor distraction (SD), the most informative feature is the 25th percentile of the subjects' BR. During the distraction segments, the subjects had an increased BR. Third, for recognizing a cognitive distraction (CD), the most informative feature is the 75th percentile of intensities of AU25 ''Lips Part''. During the distraction segments, the subjects showed increased lip movement. This could be because the cognitive-distraction sessions involved speech, which – if true – may be regarded as an artifact of the dataset rather than a general finding. Fourth, for recognizing the mixed distractions in failure session FDL, the most informative feature is the standard deviation of the activation of the emotion ''joy'', which is the same as for the ED.

Fifth, for recognizing the brake failure in session FDN, the most informative feature is the first derivative of the tonic component of nEDA. An increased positive derivative corresponds to more sweating of the subjects during the brake failure. Finally, for recognizing general distractions, the most informative feature is the difference between the minimum and the maximum values of the activation of the emotion ''joy''. This may indicate that the subjects had stronger emotional responses during the distraction segments.

B. MACHINE-LEARNING ANALYSIS

In the initial experiments, seven classical ML algorithms and seven end-to-end DL algorithms were compared for the detection of driving distraction (binary classification). The eye tracking data were not used in these experiments because the data were missing for more than 50% of the sessions. An overlapping sliding window of 20 seconds with a 5-second stride was used in these experiments. The results are presented in Table 6. Column F1 presents the F1-scores that are realized by the window classifiers, and column F1-s presents the F1-scores that are realized by the session classifiers. According to the results, the highest scores are realized by the classical ML classifiers, namely, GB and XGB. The highest F1-score for the window classifiers is 73%, and the highest F1-score for the session classifiers (column F1-s) is 88%. Among the DL classifiers, eLSTM and STRNet have similar performance, with an F1-score of 67% realized by the window classifiers.

The eLSTM session classifier realized an F1-score of 75%. and the STRNet session classifier realized an F1-score of 80%. Compared to the classical classifiers, eLSTM and STRNet outperformed the KNN, NB, DT and Bagging classifiers and were outperformed RF, GB and XGB. The experiments did not show a clear preference for the use of early or mid-fusion by the DL classifiers (denoted by the prefixes 'e' and 'm' in Table 3).

Next, a more detailed evaluation was conducted for the two best-performing classical classifiers and the two best-performing DL classifiers. Tests were conducted with various window sizes and input signals (modalities). The results are presented in Table 7. The first column presents the size of the input temporal segment in seconds (varied from 20 seconds to 80 seconds), the second column presents the ML algorithm, and the remaining columns present the F1-scores of the window classifiers (F1) and the F1-scores of the session classifiers (F1-s) for each of the input categories: face AUs, emotional activation (EMO), heart rate (HR), breathing (BR), nEDA and pEDA. For the column ''All'', all features/modalities were used as input to the classifiers. For the column ''Selected'', only the statistically significant features/modalities were used as input. According to Table 7, no classifiers perform well when only one of the physiological signals (EDA, nEDA and BR) is used as input, except the session HR classifier. The classical classifiers outperform the DL classifiers overall. Regarding the window classifiers, the highest F1 score of 79% is realized by the two classical classifiers, namely, XGB and GB, using the AUs as an input with a window size of 60 seconds. Regarding the session classifiers, the highest F1-score (F1-s) of 94% is realized by XGB using the AUs as an input with a window size of 60 seconds. Hence, the visual modalities are the most informative modalities in the experimental dataset. Among the DL classifiers, the highest performance is realized by STRNet using the selected signals and a window size of 60 seconds. The F1-score of the window classifier is 75%, and the F1-score of the session classifier (F1-s) is 87%.

Regarding the size of the input windows, all classifiers perform better with longer windows (40 seconds to 80 seconds), which is probably because longer windows contain more information.

This is especially true for the DL classifiers. Figure 4 presents the precision-recall curves of the best-performing classifiers, namely, the window classifier and the session classifier that were built with XGB using AUs as input with a window size of 60 seconds. Such curves would be useful

TABLE 7. Evaluation results for two best-performing classical classifiers and the two best-performing DL classifiers. The first column presents the size of the temporal segment in seconds. the second column presents the ML algorithm and the remaining columns present the F1-scores (%) of the window classifiers (F1) and the F1-scores (%) of the session classifiers (F1-s) for each of the specified inputs.

		All		Selected		AUs		EMO		HR		BR		nEDA		pEDA	
Win.	Classifier	F1	F1-s	F1	F1-s	F1	F1-s	F1	F1-s	F1	F1-s	F1	F1-s	F1	F1-s	F1	F1-s
20	GB	73	87	73	87	76	92	73	87	61	76	55	54	63	66	42	27
	XGB	72	88	72	88	76	90	72	82	61	76	55	65	63	63	42	27
	eLSTM	67	75	64	65	53	70	62	69	42	28	58	67	52	46	42	28
	STRNet	67	80	68	83	51	65	65	79	48	41	60	74	52	48	42	28
40	GB	77	85	77	85	79	92	75	88	64	78	57	55	65	65	41	30
	XGB	75	88	75	88	79	88	74	92	63	74	57	60	65	65	41	26
	eLSTM	70	74	68	71	54	67	66	72	55	52	59	54	63	60	42	28
	STRNet	72	82	73	87	53	61	66	77	48	41	60	74	63	64	42	28
60	GB	77	88	77	88	78	86	74	86	65	79	59	58	65	63	45	33
	XGB	76	88	76	88	**78**	**94**	74	83	66	83	59	54	65	65	45	38
	eLSTM	69	72	64	67	55	63	63	66	50	51	66	78	62	64	45	38
	STRNet	74	86	75	87	54	65	66	72	58	57	67	77	61	63	42	28
80	GB	75	92	75	92	78	92	71	82	69	83	61	64	63	65	49	33
	XGB	77	94	77	94	78	90	62	69	68	83	62	69	65	65	49	33
	eLSTM	66	71	64	67	55	61	64	70	55	52	62	69	58	59	42	33
	STRNet	72	82	73	85	56	56	66	71	58	57	69	74	60	61	42	28

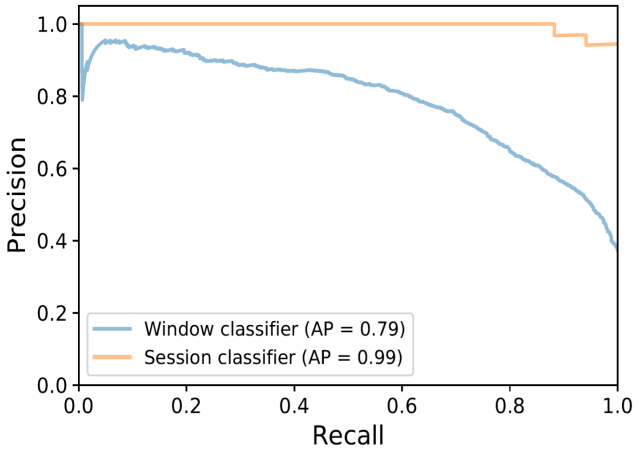

FIGURE 4. Precision-recall curves of the best-performing window classifier (blue) and session classifier (orange) that are built with XGB. AP denotes the average precision, which is defined as $\sum_n (R_n - R_{n-1}) P_n$, where R_n and P_n are the recall and precision, respectively, for the n^{th} decision threshold.

for the modification of the decision threshold. For example, in various cases, higher recall might be preferred over lower precision, as undetected distractions (false negatives) might be more dangerous than falsely detected distractions (false positives).

VI. DISCUSSION

The best-performing classical ML classifiers outperformed the best-performing DL classifiers. There may be two main reasons for this: (i) The size of the dataset is not sufficient for the end-to-end learning to outperform the best-performing classical ML classifiers. According to Table 3, the models were trained on close to 20,000 instances. While this is a large number of instances compared to related affective computing studies, which typically use a few thousand instances, it is 750 times smaller than ImageNet, which is the dataset that is used to train state-of-the-art DL NNs for image processing. (ii) DL excels in pattern recognition (e.g., image classification, object detection, and face recognition). In this use case, ''pattern recognition'', namely, emotion recognition and facial AU extraction, was conducted with other modules, and the extracted information was fed into both the classical and the DL classifiers. The access to this information probably gave the classical ML an edge as it can learn better from smaller datasets. The STRNet consistently outperformed all other classifiers when using the breathing rate (BR) as input. This is likely because spectral-domain information is especially important in relation to BR, and STRNet is the only classifier that uses time- and spectral-domain information. Classical ML models use only statistical features (except the pEDA and nEDA features), and the other DL architectures use only signals in the time domain.

The feature selection can significantly influence the classification performance of the classical feature-based ML methods. In this study, ranking-based feature selection (also known as filter methods) was used, as it is computationally efficient and does not require a classifier for feature selection. In contrast, the filter methods estimate the quality of each feature separately; hence, they fail to consider useful feature combinations. This may be the reason why the classical ML models that were built with pre-selected features did not realize the best performance. Wrapper-based feature selection methods or combinations of filter- and wrapper-based methods [71] may be useful in this case.

TABLE 8. Percentages of correctly classified instances by the best-performing window classifier and session classifier. the last row presents the accuracies of the classifiers.

Session Type	Classifier	
	Window (%)	Session (%)
ED	72	100
SD	80	100
CD	67	89
FDL	73	100
FDN	69	100
RD	93	90
ND	90	90
Overall (Accuracy)	80	95

Table 8 presents the percentages of correctly classified instances by the best-performing classifiers. The window classifier correctly classifies the windows from the normal driving sessions (ND and RD) with an average percentage of 92%, which is significantly higher than the average percentage of correctly classified windows from the distracted sessions (ED, SD, CD, FDL and FDN), which is 72%. This is probably due to the noise in the labels that is present in the windows from the distracted sessions. All windows from the normal driving sessions have the label ''normal''. However, to derive the labels of the windows from the distracted driving sessions, the following rule was used: if a distraction was present for at least 5 seconds of the window, the window should be classified as distracted, and it should be classified as normal otherwise. In various cases, the subject may need require than 5 seconds for the distraction to induce an affective response. Thus, due to the absence of an affective response, the normal windows are same as the distracted windows when analyzed using the physiological and the visual sensors. To mitigate this problem, one might use methods that explicitly incorporate label jitter into the model training process [78]. The label jitter may be why the performance on the physiological signals is worse than that on the visual signals for the detection of the driving distractions. The physiological signals may have a longer latency, namely, it may take longer for the distraction (stressor) to induce a change in the physiological signals than in the visual signals.

The session classifiers outperform the window classifiers mostly because the window classifiers must detect the exact time when the distraction occurred. In contrast, the timing is not important for the session classifiers; they need to detect only some of the distractions, which also mitigates the label-jitter problem.

For recognizing the cognitive distraction, the most informative feature was related to the driver's increased lip movement. This is expected since the cognitive-distraction sessions involved answering questions. One should be careful with using only this feature for the detection of distraction segments, as this finding may be regarded as scenario overfitting rather than a general finding. Another interesting finding is that for recognizing general distractions, the most informative feature is related to the activation of the emotion ''joy''. A more detailed analysis showed that this emotion had both higher average values and a higher standard deviation for the distracted (stressful) segments than for the normal segments. This may indicate that the normal driving sessions were more boring for the participants, whereas the driving sessions that contained distractions were more fun; as this was a driving simulation study, no distraction was regarded as dangerous by the subjects. These findings raise more general concerns regarding the generalization performances of systems that have been trained on a single dataset that was collected in a single environment. Such systems may classify any type

TABLE 9. Abbreviations.

Action Units	AUs
Adaptive Boosting	AdaBoost
Artificial Intelligence	AI
Breathing Rate	BR
Cognitive Drive	CD
Convolutional Neural Networks	CNNs
Deep Learning	DL
Electrocardiogram	ECG
Electrodermal Activity	EDA
Electroencephalogram	EEG
Electrooculogram	EOG
Emotional Activation	EMO
Emotional Drive	ED
Extreme Gradient Boosting	XGB
F1-Scores Of The Session Classifiers	F1-s
Failure-Loaded Drive	FDL
Failure-Normal Drive	FDN
Fully Connected	FC
Heart Rate	HR
Long Short-Term Memory Neural Networks	LSTMs
Machine Learning	ML
Naïve Bayes	NB
Nasal Electrodermal Activity	nEDA
Neural Network	NN
Normal Drive	ND
Palm Electrodermal Activity	pEDA
Principal Component Analysis	PCA
Relaxed Drive	RD
Sensorimotor Drive	SD
Skin Conductance Response	SCR
Spectro-Temporal Residual Network	STRNet
Support Vector Machine	SVM

of motion, speech, or emotional activation as a ''distraction'' because they were trained only for distraction detection.

In this study, all inputs were represented as 1D signals; thus, specialized DL architectures for time series were used. In the future, comparison of the methods should be made with DL classifiers that detect driver distractions directly from images. Additionally, generative adversarial networks (GANs) and transfer learning may be used to improve the performance of the DL classifiers. Furthermore, since the best-performing classifier in this study was built using AUs, in the future, different fusion strategies can be tested for the extraction of higher-level semantic facial activities (e.g., speaking, listening, and concentrating) with more semantic content.

VII. CONCLUSION

This paper presented an analysis for the determination of which ML methods perform best in detecting various driving distractions using which sensors and which data-capture methods, with a focus on physiological sensors and sensors that are based on video cameras. The statistical analysis showed that the most informative feature/modality for detecting driver distraction depends on the type of distraction. Overall, the video-based modalities were most informative, and classical ML classifiers realized high performance using one of the video-based modalities. In contrast, the DL classifiers require more modalities, namely, either all modalities or pre-selected modalities, for the construction of useful classifiers. For the analyzed data, the classical ML (XGB using the AUs as an input with a window size of 60 seconds) realized high performance and outperformed DL methods; hence, the detection of driver distractions may be technically feasible with the current knowledge. A demo of the final ML classifier is available online.[2] Finally, problems such as label jitter, scenario overfitting and unsatisfactory generalization performance were identified and discussed to provide guidance for future studies in this area.

APPENDIX

See Table 9.

REFERENCES

[1] *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*, Standard J3016, SAE International, 2018.

[2] C. Peter and R. Beale, *Affect and Emotion in Human-Computer Interaction: From Theory to Applications*. Berlin, Germany: Springer, 2008.

[3] F. Eyben, M. Wöllmer, T. Poitschke, B. Schuller, C. Blaschke, B. Färber, and N. Nguyen-Thien, ''Emotion on the road—Necessity, acceptance, and feasibility of affective computing in the car,'' *Adv. Hum.-Comput. Interact.*, vol. 2010, pp. 1–17, Jul. 2010.

[4] M. A. Regan, C. Hallett, and C. P. Gordon, ''Driver distraction and driver inattention: Definition, relationship and taxonomy,'' *Accident Anal. Prevention*, vol. 43, no. 5, pp. 1771–1781, Sep. 2011.

[5] R. J. Hanowski, R. L. Olson, J. S. Hickman, and J. Bocanegra, ''Driver distraction in commercial motor vehicle operations,'' in *Driver Distraction and Inattention*. Boca Raton, FL, USA: CRC Press, 2017, pp. 141–156.

[2]https://repo.ijs.si/martingjoreski/driving-distractions/blob/master/Demo.mp4

[6] S. Masood, A. Rai, A. Aggarwal, M. N. Doja, and M. Ahmad, ''Detecting distraction of drivers using convolutional neural network,'' *Pattern Recognit. Lett.*, early access, Jan. 12, 2018, doi: 10.1016/j.patrec.2017.12.023.

[7] I. Pavlidis, M. Dcosta, S. Taamneh, M. Manser, T. Ferris, R. Wunderlich, E. Akleman, and P. Tsiamyrtzis, ''Dissecting driver behaviors under cognitive, emotional, sensorimotor, and mixed stressors,'' *Sci. Rep.*, vol. 6, no. 1, May 2016, Art. no. 25651.

[8] J. Gomez, D. Akleman, E. Akleman, and I. Pavlidis, ''Causality effects of interventions and stressors on driving behaviors under typical conditions,'' *Mathematics*, vol. 6, no. 8, p. 139, 2018.

[9] A. Fernández, R. Usamentiaga, J. Carús, and R. Casado, ''Driver distraction using visual-based sensors and algorithms,'' *Sensors*, vol. 16, no. 11, p. 1805, 2016.

[10] B. Goel, A. Dey, P. Bharti, K. Ahmed, and S. Chellappan, ''Detecting distracted driving using a wrist-worn wearable,'' in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, Mar. 2018, pp. 233–238.

[11] B.-G. Lee and W.-Y. Chung, ''Driver alertness monitoring using fusion of facial features and bio-signals,'' *IEEE Sensors J.*, vol. 12, no. 7, pp. 2416–2422, Jul. 2012, doi: 10.1109/JSEN.2012.2190505.

[12] C.-T. Lin, Y.-C. Chen, T.-Y. Huang, T.-T. Chiu, L.-W. Ko, S.-F. Liang, H.-Y. Hsieh, S.-H. Hsu, and J.-R. Duann, ''Development of wireless brain computer interface with embedded multitask scheduling and its application on real-time driver's drowsiness detection and warning,'' *IEEE Trans. Biomed. Eng.*, vol. 55, no. 5, pp. 1582–1591, May 2008, doi: 10.1109/TBME.2008.918566.

[13] R. N. Khushaba, S. Kodagoda, S. Lal, and G. Dissanayake, ''Driver drowsiness classification using fuzzy wavelet-packet-based feature-extraction algorithm,'' *IEEE Trans. Biomed. Eng.*, vol. 58, no. 1, pp. 121–131, Jan. 2011, doi: 10.1109/TBME.2010.2077291.

[14] 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, doi: 10.1109/TITS.2005.848368.

[15] R. K. Singh, A. Sarkar, and C. S. Anoop, ''A health monitoring system using multiple non-contact ECG sensors for automotive drivers,'' in *Proc. IEEE Int. Instrum. Meas. Technol. Conf.*, May 2016, pp. 1–6, doi: 10.1109/I2MTC.2016.7520539.

[16] B. G. Lee, J.-H. Park, C. C. Pu, and W.-Y. Chung, ''Smartwatch-based driver vigilance indicator with kernel-fuzzy-C-means-wavelet method,'' *IEEE Sensors J.*, vol. 16, no. 1, pp. 242–253, Jan. 2016, doi: 10.1109/JSEN.2015.2475638.

[17] L. M. Bergasa, J. Nuevo, M. A. Sotelo, R. Barea, and M. E. Lopez, ''Real-time system for monitoring driver vigilance,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 1, pp. 63–77, Mar. 2006, doi: 10.1109/TITS.2006.869598.

[18] M. Omidyeganeh, S. Shirmohammadi, S. Abtahi, A. Khurshid, M. Farhan, J. Scharcanski, B. Hariri, D. Laroche, and L. Martel, ''Yawning detection using embedded smart cameras,'' *IEEE Trans. Instrum. Meas.*, vol. 65, no. 3, pp. 570–582, Mar. 2016, doi: 10.1109/TIM.2015.2507378.

[19] F. Vicente, Z. Huang, X. Xiong, F. De la Torre, W. Zhang, and D. Levi, ''Driver gaze tracking and eyes off the road detection system,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 2014–2027, Aug. 2015, doi: 10.1109/TITS.2015.2396031.

[20] E. Murphy-Chutorian and M. M. Trivedi, ''Head pose estimation and augmented reality tracking: An integrated system and evaluation for monitoring driver awareness,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 2, pp. 300–311, Jun. 2010, doi: 10.1109/TITS.2010.2044241.

[21] 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, doi: 10.1109/TITS.2003.821342.

[22] C. Yan, F. Coenen, and B. Zhang, ''Driving posture recognition by convolutional neural networks,'' *IET Comput. Vis.*, vol. 10, no. 2, pp. 103–114, Mar. 2016, doi: 10.1049/iet-cvi.2015.0175.

[23] B.-L. Lee, B.-G. Lee, and W.-Y. Chung, ''Standalone wearable driver drowsiness detection system in a smartwatch,'' *IEEE Sensors J.*, vol. 16, no. 13, pp. 5444–5451, Jul. 2016, doi: 10.1109/JSEN.2016.2566667.

[24] A. Aksjonov, P. Nedoma, V. Vodovozov, E. Petlenkov, and M. Herrmann, ''Detection and evaluation of driver distraction using machine learning and fuzzy logic,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 6, pp. 2048–2059, Jun. 2019.

[25] Y. Saito, M. Itoh, and T. Inagaki, ''Driver assistance system with a dual control scheme: Effectiveness of identifying driver drowsiness and preventing lane departure accidents,'' *IEEE Trans. Human-Machine Syst.*, vol. 46, no. 5, pp. 660–671, Oct. 2016, doi: 10.1109/THMS.2016.2549032.

[26] N. Apostoloff and A. Zelinsky, "Vision in and out of vehicles: Integrated driver and road scene monitoring," *Int. J. Robot. Res.*, vol. 23, nos. 4–5, pp. 513–538, 2004.
[27] G. Castignani, T. Derrmann, R. Frank, and T. Engel, "Driver behavior profiling using smartphones: A low-cost platform for driver monitoring," *IEEE Intell. Transp. Syst. Mag.*, vol. 7, no. 1, pp. 91–102, Apr. 2015, doi: 10.1109/MITS.2014.2328673.
[28] J. Wang, C. Yu, S. E. Li, and L. Wang, "A forward collision warning algorithm with adaptation to driver behaviors," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 1157–1167, Apr. 2016, doi: 10.1109/TITS.2015.2499838.
[29] G. Sikander and S. Anwar, "Driver fatigue detection systems: A review," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 6, pp. 2339–2352, Jun. 2019.
[30] A. Borbély, "A two process model of sleep regulation," *Hum. Neurobiol.*, vol. 1, no. 3, pp. 195–204, 1982.
[31] A. J. Belyavin and M. B. Spencer, "Modeling performance and alertness: The QinetiQ approach," *Aviation, Space Environ. Med.*, vol. 75, no. 3, pp. A93–A103, 2004.
[32] T. Azim, M. A. Jaffar, and A. M. Mirza, "Fully automated real time fatigue detection of drivers through fuzzy expert systems," *Appl. Soft Comput.*, vol. 18, pp. 25–38, May 2014.
[33] A. Koesdwiady, S. M. Bedawi, C. Ou, and F. Karray, "End-to-end deep learning for driver distraction recognition," in *Proc. Int. Conf. Image Anal. Recognit.* Cham, Switzerland: Springer, 2017, pp. 11–18.
[34] M. S. Majdi, S. Ram, J. T. Gill, and J. J. Rodriguez, "Drive-net: Convolutional network for driver distraction detection," in *Proc. IEEE Southwest Symp. Image Anal. Interpretation (SSIAI)*, Apr. 2018, pp. 1–4
[35] M. Hssayeni, S. Saxena, R. Ptucha, and A. Savakis, "Distracted driver detection: Deep learning vs handcrafted features," *Electron. Imag.*, vol. 2017, no. 10, pp. 20–26, Jan. 2017.
[36] Y. Zhou, P. Tsiamyrtzis, P. Lindner, I. Timofeyev, and I. Pavlidis, "Spatiotemporal smoothing as a basis for facial tissue tracking in thermal imaging," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 5, pp. 1280–1289, May 2013.
[37] D. Shastri, M. Papadakis, P. Tsiamyrtzis, B. Bass, and I. Pavlidis, "Perinasal imaging of physiological stress and its affective potential," *IEEE Trans. Affect. Comput.*, vol. 3, no. 3, pp. 366–378, Jul. 2012.
[38] T. Hassan, D. Seuss, J. Wollenberg, J. Garbas, and U. Schmid, "A practical approach to fuse shape and appearance information in a Gaussian facial action estimation framework," in *Proc. 22nd Eur. Conf. Artif. Intell., 9th Int. Conf. Prestigious Appl. Artif. Intell. (PAIS)*, vol. 285, G. A. Kaminka, Ed., 2016, pp. 1812–1817.
[39] P. Ekman and W. V. Friesen, *Facial Action Coding System*. Palo Alto, CA, USA: Consulting Psychologists Press, 1978.
[40] P. Ekman, W. V. Friesen, and J. C. Hager, *The Facial Action Coding System*, 2nd ed. Salt Lake City, UT, USA: Research Nexus eBook, 2002.
[41] J. F. Cohn, Z. Ambadar, and P. Ekman, "Observer-based measurement of facial expression with the facial action coding system," in *Handbook of Emotion Elicitation and Assessment* (Series in Affective Science), J. A. Coan and J. J. B. Allen, Eds. London, U.K.: Oxford Univ. Press, 2007, pp. 203–221.
[42] M. Gjoreski, M. Luštrek, M. Gams, and H. Gjoreski, "Monitoring stress with a wrist device using context," *J. Biomed. Informat.*, vol. 73, pp. 159–170, Sep. 2017.
[43] *Scikit-Learn*. [Online]. Available: https://scikit-learn.org/stable/
[44] J. R. Quinlan, "Improved use of continuous attributes in C4.5," *J. Artif. Intell. Res.*, vol. 4, pp. 77–90, Mar. 1996.
[45] T. K. Ho, "Random decision forests," in *Proc. 3rd Int. Conf. Document Anal. Recognit.*, vol. 1, Aug. 1995, pp. 278–282.
[46] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Kuala Lumpur, Malaysia: Pearson Education Limited, 2016.
[47] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Mach. Learn.*, vol. 6, no. 1, pp. 37–66, Jan. 1991.
[48] J. Shawe-Taylor and N. Cristianini, *An Introduction to Support Vector Machines: And Other Kernel-Based Learning Methods*, vol. 204. Cambridge, U.K.: Cambridge Univ. Press, 2000.
[49] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, Aug. 1996.
[50] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *Proc. Eur. Conf. Comput. Learn. Theory*. Berlin, Germany: Springer, 1995, pp. 23–37.
[51] L. Deng, "Deep learning: Methods and applications," *Found. Trends Signal Process.*, vol. 7, nos. 3–4, pp. 197–387, 2014, doi: 10.1561/2000000039.
[52] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bull. Math. Biophys.*, vol. 5, no. 4, pp. 115–133, Dec. 1943, doi: 10.1007/BF02478259.
[53] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 1–8.
[54] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural Netw.*, vol. 12, no. 1, pp. 145–151, Jan. 1999.
[55] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[56] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[57] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proc. 26th Annu. Int. Conf. Mach. Learn. (ICML)*, 2009, pp. 609–616.
[58] K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?" in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 2146–2153.
[59] X. Jiang, Y. Pang, X. Li, and J. Pan, "Speed up deep neural network based pedestrian detection by sharing features across multi-scale models," *Neurocomputing*, vol. 185, pp. 163–170, Apr. 2016.
[60] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1–7.
[61] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *IEEE Comput. Intell. Mag.*, vol. 13, no. 3, pp. 55–75, Aug. 2018.
[62] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, Feb. 2003.
[63] S. Chaib, H. Yao, Y. Gu, and M. Amrani, "Deep feature extraction and combination for remote sensing image classification based on pre-trained CNN models," in *Proc. 9th Int. Conf. Digit. Image Process. (ICDIP)*, Jul. 2017, Art. no. 104203.
[64] T. Vandal, E. Kodra, S. Ganguly, A. Michaelis, R. Nemani and A. R. Ganguly, "DeepSD: Generating high resolution climate change projections through single image super-resolution," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2017, pp. 1663–1672.
[65] C. Chen, Q. Chen, J. Xu, and V. Koltun, "Learning to see in the dark," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3291–3300.
[66] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*. [Online]. Available: http://arxiv.org/abs/1810.04805
[67] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[68] F. A. Gers and J. Schmidhuber, "Recurrent nets that time and count," in *Proc. IEEE-INNS-ENNS Int. Joint Conf. Neural Netw. (IJCNN)*, vol. 3, Jul. 2000, pp. 189–194.
[69] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural Comput.*, vol. 12, no. 10, pp. 2451–2471, Oct. 2000, doi: 10.1162/089976600300015015.
[70] K. Cho, B. van Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, "Learning phrase representations using RNN encoder-decoder," in *Proc. Conf. Empirical Methods Natural Lang. Process. Stat. Mach. Transl.*, 2014, pp. 1724–1734.
[71] M. Gjoreski, V. Janko, G. Slapnicar, M. Mlakar, N. Rescic, J. Bizjak, V. Drobnic, M. Marinko, N. Mlakar, M. Lustrek, and M. Gams, "Classical and deep learning methods for recognizing human activities and modes of transportation with smartphone sensors," *Inf. Fusion*, 2020.
[72] M. Gjoreski, A. Gradisek, B. Budna, M. Gams, and G. Poglajen, "Machine learning and end-to-end deep learning for the detection of chronic heart failure from heart sounds," *IEEE Access*, vol. 8, pp. 20313–20324, 2020, doi: 10.1109/ACCESS.2020.2968900.
[73] G. Slapničar, N. Mlakar, and M. Luštrek, "Blood pressure estimation from photoplethysmogram using a spectro-temporal deep neural network," *Sensors*, vol. 19, no. 15, p. 3420, 2019.
[74] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[75] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*. [Online]. Available: http://arxiv.org/abs/1502.03167
[76] F. Wilcoxon, "Individual comparisons by ranking methods," in *Breakthroughs in Statistics*. New York, NY, USA: Springer, 1992, pp. 196–202.
[77] H. Hofmann, H. Wickham, and K. Kafadar, "Value plots: Boxplots for large data," *J. Comput. Graph. Statist.*, vol. 26, no. 3, pp. 469–477, 2017.
[78] H. Kwon, G. D. Abowd, and T. Plötz, "Handling annotation uncertainty in human activity recognition," in *Proc. 23rd Int. Symp. Wearable Comput. (ISWC)*, 2019, pp. 109–117.

MARTIN GJORESKI received the B.S. degree in computer science from the Faculty of Computer Science and Engineering, Skopje, Macedonia, in 2014, and the M.S. degree in computer science from the Jožef Stefan Postgraduate School, Ljubljana, Slovenia, in 2016, where he is currently pursuing the Ph.D. degree in computer science. He has been a Research Assistant with the Department of Intelligent Systems, Jožef Stefan Institute, Ljubljana, since 2014. His research interest includes development of machine-learning methods for monitoring human states using wearable sensors. He developed parts of the machine-learning algorithms that received the Sussex-Huawei Locomotion Challenge, in 2018 and 2019, respectively.

MATJAŽ GAMS (Member, IEEE) received the Ph.D. degree. He is currently the Head of the Department of Intelligent Systems, Jožef Stefan Institute, Ljubljana, and a Professor of computer science with the University of Ljubljana and the Jožef Stefan Postgraduate School. His research interests include intelligent systems, artificial intelligence, cognitive science, intelligent agents, electronic and mobile health, business intelligence, and information society. He is a member of several international program committees of scientific meetings, national, the European Strategic Boards and Institutions, and editorial boards of 11 journals. He is also a member of the National Council of Slovenia, representing the field of science, for a period of 2017 to 2022. His team received four activity recognition competitions and placed in the finals of the XPrize Tricorder Competition. He is the Managing Director of journal *Informatica*.

MITJA LUŠTREK (Member, IEEE) received the Ph.D. degree in computer and information science from the University of Ljubljana. He is currently the Head of the Ambient Intelligence Group, Department of Intelligent Systems, Jožef Stefan Institute, Slovenia. His research interests include applying machine learning to sensor and other data in the domains of health, wellbeing, and ambient assisted living. His team received four activity recognition competitions and placed in the finals of the XPrize Tricorder Competition.

PELIN GENC received the B.Sc. and M.Sc. degrees in mechanical engineering from Middle East Technical University, Ankara, Turkey. She is currently pursuing the master's degree in informatics with Friedrich-Alexander University Erlangen-Nuremberg, Erlangen. From 2017 to 2019, she was a Research Assistant with Fraunhofer IIS. Her research interests include multimodal stress recognition, with a special focus on facial expressions, physiological signals, and psychophysiology of emotions and stress.

JENS-U. GARBAS received the Diploma and Ph.D. degrees in electrical engineering from Friedrich-Alexander-University Erlangen-Nuremberg, in 2004 and 2010, respectively.

In 2010 he joined Fraunhofer IIS. He was the Head of the Intelligent Systems Group, in 2011, and the Deputy Head of the Electronic Imaging Department, in 2012. Since 2020, he has been the Head of the Department Digital Sensory Perception. He is responsible for industrial and public research projects and software licensing in the areas of real-time computer vision, affective computing, and facial analysis.

TEENA HASSAN received the Bachelor of Technology degree in computer science and engineering from the Cochin University of Science and Technology, Kerala, India, and the Master of Science degree in autonomous systems from the Bonn-Rhein-Sieg University of Applied Sciences, Sankt Augustin, Germany, in 2014. She is currently pursuing the external Ph.D. degree with the University of Bamberg. She continued her research in the field of automatic facial expression analysis with Fraunhofer IIS, Erlangen, Germany. She is currently a Research Associate with Bielefeld University and working in interaction architectures for social robots. Her research interests include modeling of facial muscle movements, fusing multiple sources of facial expression information, and modeling uncertainty in observations. Her contributions in this article are part of her research with Fraunhofer IIS.