AI-Based Driving Data Analysis for Behavior Recognition in Vehicle Cabin

Friedrich Lindow
University of Rostock, Rostock, Germany
ITMO University, St. Petersburg, Russia
friedrich.lindow@uni-rostock.de

Christian Kaiser
Virtual Vehicle Research GmbH, Graz, Austria
University of Rostock, Rostock, Germany
christian.kaiser@v2c2.at

Alexey Kashevnik
ITMO University, St. Petersburg, Russia
SPIIRAS, St. Petersburg, Russia
alexey@iias.spb.su

Alexander Stocker
Virtual Vehicle Research GmbH, Graz, Austria
alexander.stocker@v2c2.at

Abstract—**Driving a vehicle is an indispensable part of their everyday life for many people. However, sometimes this everyday life does not go as expected, as a lot of accidents happen on the public roads, and most of these accidents are due to inattentive driver behavior. Modern driver monitoring systems evaluate driver behavior by means of distinctive sensor technology and, if necessary, indicate undesirable driving behavior. However, many roadworthy vehicles do not have the possibility to implement such systems. Therefore, it seems to be interesting to investigate the implementation of such systems based on commodity hardware, e.g., smartphones, because nowadays almost every driver has a powerful smartphone equipped with many sensors at hand in the vehicle. Furthermore, recent advances in Machine Learning (ML) made it possible to analyze large amounts of data and to generate new outcomes. In this work we discuss how ML can be used for driver behavior recognition by improving an already existing threshold-based driver monitoring system with different ML-based techniques, Neural Networks and Random Forests, and evaluate their performance. We propose to use Microsoft Azure platform to analyze data generated by a Driver Monitoring System (DMS). Our results indicate ML as a useful technique for learning and adapting threshold-based reasoning about individual drivers' states.**

I. INTRODUCTION

The task of driving is coupled with the everyday life of many people around the globe. Being for a means of transportation or in the context of everyday work, driving can be seen as omnipresent in our society. This raises special concerns when it comes to safety, because driving is dangerous and can lead to fatalities. According to the statistics of traffic fatalities for 2018 in the United States, a total of 36,560 people died in motor vehicle traffic crashes [1]. Worldwide, the number of people killed totals up to 1.35 million per year. These figures make road traffic fatalities the eighth largest cause of death for people of all ages [2].

The societal desire to reduce the number of deaths leads to increased funding, research, and opportunities to improve the given situation. Hence, making traffic of the future safer leads to less crashes and accidents und ultimately safes lives, as it is well-known from related work, that the driving behavior is a critical role for traffic safety [3–5]. Analyzing and evaluating the behavior of drivers takes a mandatory role for the newly emerging opportunities within the transportation domain, which is driven by the digitalization. This latter includes not only new ways of dealing with data, but also the sheer amount of data that is being produced and captured. Ultimately, it is possible because there is a rise in vehicle sensors, which leads to new data about vehicles and driving performance [6]. This development is further enhanced by the upcoming use of Cloud technologies and along with advanced ways of analyzing, communicating, and storing data. These domain opportunities can be tackled with a wide range of methods and algorithms.

An upcoming field, which seems promising for the task of driving behavior analysis, is the Artificial Intelligence (AI) domain. With the increase in the availability of driver's data, their driving styles, and their trips, enhanced conclusions can be made about how the identified driver behavior is related to *safe* and *unsafe* driving. Thus, identifying the driving behavior can lead to an increase in overall traffic safety. In order to achieve this goal, the evaluated driver's behavior should be used to warn each individual driver (and probably his/her environment, as well) about his/her condition and give reasonable recommendations towards getting the desired safe way of driving [7].

Another topic that should be mentioned is the accessibility of such systems. It is only in recent years that vehicle manufacturers have cautiously begun to integrate interfaces for data into the vehicles. A range of sensors (e.g., cameras detecting the driver status) and interfaces are only available in certain modern vehicles, with the number of sensors will continue to steadily increase, as an example when considering automated driving functions (e.g., RADAR and LiDAR sensors). This leads to a mix of differently equipped vehicles on the road. Sadly, owners of older vehicles cannot use innovations based on vehicle's sensor data at all. Due to this, it would be purposeful to decouple the vehicle and the data acquisition infrastructure. A clever way of doing that is to take what is already in use, e.g., commodity hardware. Hence, a smartphone, which is widely popular and accessible, can be used for this task, since it combines a wide range of sensors

ISSN 2305-7254

applicable for this task, and provides the communication infrastructure for sharing data [8].

Based on smartphone data, this paper introduces an extended architecture model for a given threshold-based driver behavior detection system, denoted as "Drive Safely" [9]. It incorporates the use of Machine Learning (ML) techniques for learning and adapting the threshold-based reasoning about the individual driver´s state. For this study, the following research questions (RQs) are considered:

RQ1: How can ML techniques be implemented for driver behavior recognition?

RQ2: What are the most relevant features for driver behavior recognition?

The paper is organised as follows. In Section II related works are presented, while Section III describes the architecture model. Section IV explains data preparation steps, followed by the proposed ML-oriented model in Section V. Experimental results regarding the implementation of the proposed AI-based technique is presented in Section VI, followed by a discussion in Section VII. Finally, in Section VIII conclusions are provided.

II. Related Work

In the following, some scientific findings which have already been published in order to be able to base our research on them, are analysed. In particular, a literature review was carried out, regarding the usage of smartphone sensor data and ML methods for the identification of the driver behavior [7]. In the following we summarize ten relevant papers we found on scientific databases of SCOPUS and ScienceDirect, which are selected through inclusion and exclusion criteria.

The work of Ferreira et al. considers driver classification with different Android-based smartphone sensors [10]. In detail, they investigate which sensor and method assembly return the highest performance: accelerometer, linear acceleration, magnetometer, and gyroscope are smartphone sensors, and the classification task is done with different ML algorithms like Artificial Neural Networks (ANNs), Support Vector Machines (SVMs), Random Forest, and Bayesian Network. The results show that Random Forest ranks first followed by ANNs. Furthermore, authors in [10] investigate how the sliding window size for data accumulation influences the performance. To this end, they showed, that larger time windows perform better, and that gyroscope and accelerometer are the best sensors for their classification task.

In [4], an investigation of driving skill classification by analyzing the skill at maneuvering in curve driving scenes, is presented. ML-oriented methods are used to classify the driver according to these driving capabilities, with the overall aim to create a framework in order to make driving more comfortable and safer. The driver model is trained from sensor data gathered from a driving simulation related to the driving environment, vehicle response, and driving behavior for driver. Once the model is trained, novel driving situations can be classified automatically. Moreover, in [4] the k-nearest neighbor (k-NN) classifier and SVM are used, thus taking different curve driving scenes as the basis of the classification in order to tag the drivers in different driving skill levels. Then, relevant features include: steering angle, speed, longitudinal acceleration, lateral acceleration, yaw rate, accelerator control, brake control, lateral displacement, longitudinal displacement, accelerator control speed, and brake control speed.

TABLE I. LITERATURE REVIEW SEARCH QUERIES.

Search Query
"machine learning" AND ("driving behaviour" or "driving behavior") AND (smartphone OR mobile) AND (safety OR accident)
"driver classification" AND (behaviour OR behavior) AND "machine learning"
"driver behavior" AND classification AND "machine learning"

Finally, authors in [4] conclude that SVM performs better than k-NN algorithm for the chosen mode, since for the classification in full curve scenes, SVM reaches an accuracy of 95.7%, while for classification in segmented curves, the average accuracy scored 89%.

A framework for delivering personalized and quantified driving policies implemented with the use of smartphone sensors data is shown in [10], Where Q-learning is used as a representative of reinforcement learning techniques. The inputs for the Q-learning algorithm consist in the number of harsh events per kilometer, speeding and mobile usage, which are derived from the motion and position sensors such as accelerometer, gyroscope and magnetometer. Their findings confirm the necessity of a personalized approach of quantifying driver behavior.

In [11] the driving behavior with regard to the respective air pollutants emissions, is investigated. In detail, authors use a K-means clustering algorithm on a large database of 4156 trips to cluster those trips in three profiles, namely driving *as usual*, *harsh* driving, and *eco-friendly* driving. In this case, the smartphone acts as an embedded sensor platform for this approach and the accelerometer, gyroscope and magnetometer represent the data sources.

Authors in [12] study anomaly recognition algorithms to detect aggressive driving. In detail, they use vehicle motion data collected from smartphone sensors (such as accelerometer and gyroscope) and analyze data with Gaussian mixture model, partial least squares regression, wavelet transformation and Support Vector Regression (SVR). They conclude that these algorithms can recognize aggressive driving behavior, with Gaussian mixture model and SVR being superior with respect to the other two algorithms. The anomaly detection with the help of acceleration data is more recognizable than those derived from the gyroscope data. Finally, they find out that small changes to the range of thresholds does not impact the results which leads to the finding, that correctly selected threshold values can still come with small changes in the environment and consequently the performance of the algorithm.

In [13], driving behavior analysis is performed. In detail, asystematic literature review on the topic with a special emphasis on ML approaches is presented, thus identifying an interpretive framework incorporating multiple dimensions influencing the driver's behavior and summarizing the available evidence of the ML techniques for constructing driver behavior estimation models. Furthermore, authors in [13] compare ML approaches with non-ML models and the accuracy between the ML models as well. They conclude that ML models outperform non-ML models in general. Additionally, they give recommendation for researchers to carry out future research on driver behavior analysis using ML techniques.

In [14], the possibility to detect and predict an impaired driver state, such as drowsiness, by developing an ANN for those tasks, is investigated. Furthermore, the information provided to the algorithm is collected from different data sources ranging from physiological, behavioral, to psychological data about the driver, as well as performance information from the vehicle. Different datasets from different data sources were evaluated in order to determine the most optimal success in detecting and predicting impairment. Data are collected from participants in a driving simulator. The generalization and inter-individual variability is viewed as a challenging task in order to evaluate drivers whose data is not trained beforehand.

Authors in [15] proposed a model for detecting sudden braking and aggressive driving behaviors with data collected from smartphone sensors. A dynamic time warping technique is used for classification purposes in mobile devices with constrained resources. The proposed algorithm has an accuracy of 100% for detecting braking events, 97% for detecting left and right turns, and 86.67% for detecting aggressive turns.

In [16], a personalized driving state recognition system, taking into account not only personalized driving characteristics, but also considering contextual information, such as the road type, is proposed. This leads to an improvement in accuracy of the driving state recognition for individual drivers. As the classification algorithm, authors compare discriminant analysis, decision tree, k-NN, SVM and Random Forest, and they found that the latter outperforms the other techniques.

Authors in [17] proposed a fine-grained abnormal driving behavior detection and identification system using smartphone sensors (in particular acceleration and orientation) in order to train both a SVM and neuron network with empirically grounded data from real driving situations. The identification of abnormal driving events distinguishes between weaving, swerving, side slipping, fast U-turns, wide-radius turning and sudden breaking. Furthermore, they investigate different impacts on the results such as training set size, traffic condition, road type, smartphone placement and the sensors sampling rate.

The reviewed papers face challenges about the implementation and configuration of ML algorithms for driver behavior recognition for driver of vehicles. They propose different algorithms. The most promising algorithms are identified in [7] and consequently the neural network and random forest algorithms are chosen to be compared against each other.

Based on the findings from preliminary works, in the following the proposed ML algorithm is presented and discussed. In order to do this, we collect data from the smartphone sensors, feed them into the ML-oriented infrastructure and derive results. The latter shall be executed in a Cloud service. In the next section we will go into more detail about the architecture model.

III. Architecture

The architecture of the driver monitoring system (Fig. 1) derives from the aim of this paper to analyze and classify driving data for behavior recognition. In detail, there are two distinctive analyzing components: 1) a mobile application analyzing the data live on the smartphone via thresholds, and 2) a Cloud service to learn from user feedback and update thresholds.

For the application analyzing the data live, a threshold-based detection scheme running on the smartphone is chosen. It analyses smartphone sensor data and creates events when thresholds are exceeded. As humans behave individually, we use the data analysis service running in the Cloud to update the thresholds for each individual user. This adjustment is based on the sensor data and feedback from the user, e.g., if the user confirms or denies the event that says he was driving drowsy. For this classification task, different ML techniques are utilized.

In more detail, the procedure is the following. Regarding the mobile app running inside the smartphone, in order to collect data, the driver places the smartphone in his/her driver cabin and starts the logging application. While driving, the smartphone application collects smartphone sensor data about the driving behavior of the driver. The threshold-based detection system performs both online and offline a *dangerous state detection* and, according to these thresholds, different dangerous states can be identified. Based on that, the recommendation module raises a suitable alert for the driver. Additionally, recorded data are uploaded to the Cloud in order to perform the data analysis (as shown in Fig. 1). The ML component acts as an enhancement to the threshold-based system, which raises alerts, based upon predefined thresholds. This is done by taking confirmation of the dangerous events by the driver into account. This way, the ML classification gains extended knowledge about abnormal driving to update the current thresholds of the drive safety system iteratively. Different driving styles and overall driving behavior might be perceived differently for varying drivers. If each driver labels its recorded data as correct or incorrect and proposed behavior identification, then the system should find better fitting thresholds for the identification of distraction and drowsiness.

Regarding the Cloud-oriented component, the corresponding sensor data is sent to the Cloud CSV-formatted for further analysis. Within the ML component, stored data are prepared for the purpose of the following classification task, with the help of a Python script. A detailed description of the data preparation is provided in Section IV. Prepped data are fed to the classification model of Microsoft Azure, which performs the classification of the labeled events. A more detailed view on the classification model is provided in Section V. According to deviations to the existing thresholds, the particular driver gets assigned to a driver group which is more fitting to his provided feedback of the detected dangerous situations. Based on those driver groups, the personalized driver preferences are derived concerning the thresholds for the individual driver.

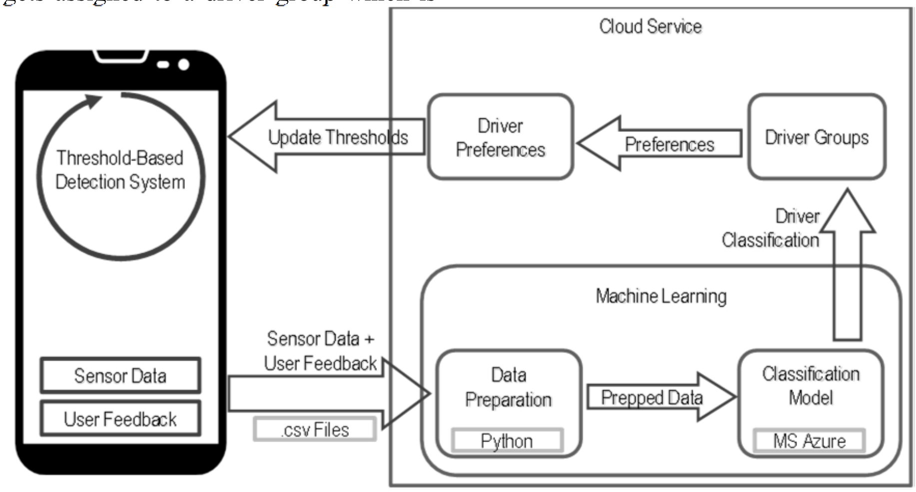

Fig. 1. Architecture of the driver monitoring system

Afterwards, the updated thresholds are sent to the driver's smartphone and will replace the old thresholds for a more personalized detection scheme, so far this is not considered to update the thresholds live, because the user gives his/her feedback (accept or deny dangerous states) after a recorded trip. The main area of work for this paper is the ML component with the data preparation and the classification model. The processes behind the driver grouping, personal driver preferences, and the updating of the threshold-based detection system is out of the scope of this paper.

IV. Data Preparation

A. Acquisition

The raw data was recorded using an already existing threshold-based system for the driving style recognition [18]. Then the driving style of the participants was recorded and labeled in the same way these thresholds have been made previously labeled. Ultimately, the domain experts in this field, i.e., the drivers should continue to give feedback to the findings by labelling the recognized critical data themselves, how they perceived their driving style and therefore confirm or correct the label for the particular event.

B. Raw Data

The raw data comes in two datasets, one containing the pure records of all the trips except the labels, called *pure,* and the other dataset contains the recognized critical driving data, called *critical*.

Both datasets contain extensive information about the driving events. The necessary data for the classification task do including all the miscellaneous data which can be found in those source datasets. In order to have one dataset which includes all the necessary data, a join task needs to be conducted. For a successful join with *critical* and *pure* data, the timestamps in both datasets are extracted, thus serving as primary keys for the join task. In the case of the *pure* dataset each row represents a snapshot of that time, the *critical* dataset sums up all driving snapshots which are related to a particular recognized driving style event, e.g., drowsy driving. The new dataset contains, after the join task, all data from all critical events such as distracted or drowsy driving with their respective label, and the associated feature columns.

C. Data Cleaning

The raw data was not uniformly, having some missing values or mislabeled columns which required it to be cleaned. Consequently, all rows which contain missing values got dropped. This applies to NaN values but also to those where the feature columns are all equal to zero. Hence, this implies a recording error for that corresponding event and therefore it gets dropped as well. This means in practice that the face of the driver was not in the target area of the camera and could not be measured.

D. Data Selection

The joined dataset contains the label column, i.e., the dangerous driving behavior state, as well as the features that are important for the classification of the driving events. The head angle, the mouth openness ratio, as well as the eye openness are considered to be important in the threshold-based detection system for both drowsiness and distraction detection [18]. In order to cope with the existing threshold-based system concerning comparability, the respective columns have been chosen as features. Consequently, features about the openness of both eyes individually, as well as information about the head concerning the yaw and pitch angle and lastly, the openness ratio of the mouth while driving are considered. This dataset builds the ground knowledge for the classification task. Within the ML model, some additional data are added for evaluating the results; these are newly gathered information by the threshold-based system from a newly recorded trip. It gets alighted to the shape of the joined dataset, which means it contains the same columns. This is done to simulate the real-world use case, where an already trained ML algorithm takes new data from a new trip and performs its classification task accordingly.

The described feature columns represent the data, which are used to determine the dangerous state in the threshold-based system, described in [18] for various dangerous states, and, according to that approach, the corresponding features are used in the ML classification as well. Besides those features the ML algorithms can take additional data as input. It is tested how the classification task improves when additional data is added as features, this results in different data configurations and should ultimately lead to a recommendation on what data should be used in order to improve classification results. The underlying threshold-based detection system detects dangerous states based upon a predefined time window of 1,5 seconds. A similar approach is a sliding time window concept used to aggregate features. Consequently, for each event in the dataset, a time window detects every event which is relevant and performs calculations on them. The following table shows the different baselines (data configurations), which will be compared using the Neural Network and Random Forest ML approaches. We use these four baseline variations to find the best min required events as well as time window.

The proposed classification model (Fig. 2) is implemented through Microsoft Azure designer, since it provides a stable platform and offers a suitable testing ground, with the possibility to deploy models for the real world use case as well. The model is organized in logical modules and they may or may not be depend on another, the dependency is modelled via the arrows, which link the corresponding modules.

All starts with the import of the dataset, which is firstly cleaned in the data preparation module, in the depicted case the Baseline 3 dataset is given as an example. It contains both the target-labels and the features for every entry. The next module, Clean Missing Data, is a preconstructed Microsoft Azure Module. Its whole purpose is to have error resistant data by performing one last data cleaning process on the source data, to ensure its compatibility with the Microsoft Azure environment by sorting out any data that is, in any way, not applicable to the following classification task.

In the upper left corner of the model is the ML algorithm located. This particular model represents a Multiclass Neural Network classification module, a promising ML approach derived from the literate review [7]. Other ML algorithms like Random Forests take the place of the Multiclass Neural Network-module when they are tested. There are two separate modules, because afterwards two classification tasks are performed, both taking a single ML algorithm module as input. The first classification task, which is positioned on the left side of the model, is a Cross Validate Model. It is used to access both the variability of a dataset and the reliability of a model. This is done by dividing the dataset into K-folds and performing K classification tasks: each one takes a different fold as the validation set. In the end, the evaluation metrics are performed on the averages of the K folds in order to test the model with different train and testing data each time. This leads to stable and less variable results. The evaluation is done by the Evaluate Model-module and evaluation metrics such as Accuracy, Precision, F1-score, AUC and Recall are given. The second classification task mimics the appearance of new and unseen data. In reality this can be a new trip or a new driver using the application. This new data is consequently used as a validation set. In order to bring both the source dataset, in this case Baseline 3 and the validation set Scoring Dataset in a compatible shape, both datasets are passed to an Edit Metadata-module where the datatype of the columns are set uniformly.

TABLE II. DATA CONFIGURATION

Baseline	Features	Feature-Aggregations	Min Req. Events	Time Window
1	Eye openness, head angle pitch, head angle yaw, mouth openness, speed	custom PERCLOS, mean eye openness, max eye openness, min eye openness, mean head angle pitch, mean head angle yaw, min head angle pitch, min head angle yaw, max head angle yaw, max head angle pitch	10	1,5 sec
2	Eye openness, head angle pitch, head angle yaw, mouth openness, speed	custom PERCLOS, mean eye openness, max eye openness, min eye openness, mean head angle pitch, mean head angle yaw, min head angle pitch, min head angle yaw, max head angle yaw, max head angle pitch	14	1,7 sec
3	Eye openness, head angle pitch, head angle yaw, mouth openness, speed	custom PERCLOS, mean eye openness, max eye openness, min eye openness, mean head angle pitch, mean head angle yaw, min head angle pitch, min head angle yaw, max head angle yaw, max head angle pitch, mean mouth openness, min mouth openness, max mouth openness	10	1,5 sec
4	Eye openness, head angle pitch, head angle yaw, mouth openness, speed	custom PERCLOS, mean eye openness, max eye openness, min eye openness, mean head angle pitch, mean head angle yaw, min head angle pitch, min head angle yaw, max head angle yaw, max head angle pitch, mean mouth openness, min mouth openness, max mouth openness	12	1,5 sec

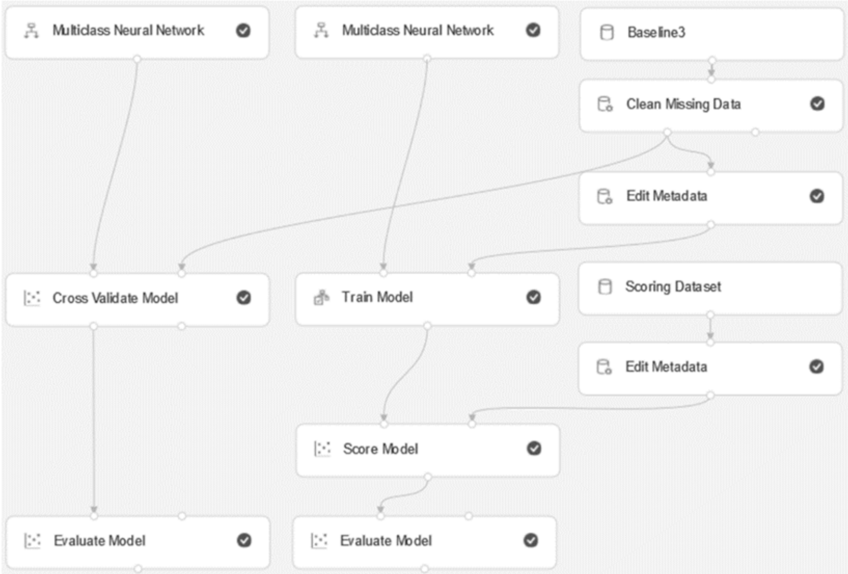

Fig. 2. The proposed ML model

The training set Baseline 3 is then passed to the module Train Model and the validation set gets passed to the Score Model Task. The Train Model also takes a ML algorithm as input. The model is trained based on the Training-Data using the exemplary Neural Network and afterwards the trained Neural Network will be passed to the Score Model Task where it gets tested on the validation set which contains the new data. In the last step, the model gets evaluated based on Accuracy, Precision, F1-score, AUC and Recall.

VI. Experimental results

A comparative approach between different ML techniques allows to choose the most suitable algorithm. In [19] the concern that ML experiments rely heavily on the data is raised, the interpretation if it and are hard to reproduce which is of special interest in the research field. Previous researchers showed that for similar classification tasks, Neural Networks, Random Forest, and SVM are widely used, these are consequently chosen as algorithms to compare against each other [7]. The Microsoft Azure classification model has some parameters for tweaking the ML algorithms. Each different algorithm has its own parameters. For the Neural Network the number of hidden nodes in the Neural Network, as well as the learning rate and the number of iterations, are taken into account when optimizing the algorithm. Initial testing revealed that 100 hidden nodes perform well, lowering the number of hidden nodes decreases performance, and with an increase above 100 nodes that does not have a big impact on the results. Consequently, that parameter is fixed to 100 nodes. The parameter for the SVM is the number of iterations and lambda, which denotes the degree of importance for misclassifications, with bigger lambdas the misclassifications are more important and less allowed. The Random Forest has the number of individual trees, their maximum depth, and the minimal number of samples per leaf node as parameter. Each model parameter configuration is attached to one evaluation row.

With reference to Table II, the described data configurations resemble the most prevalent use case of the proposed system which can detect and distinguish multiple different dangerous states, such as distraction and drowsiness at once. Configuration 1 and 2 include, as features, the custom perclos, the speed, the openness of both the left and the right eye, the head pitch, the head yaw and the openness of the mouth. Furthermore, additional features are the mean, the minimum and maximum values of the eyes, the head yaw and the head pitch. The configuration 3 and 4 extend the three described configurations with aggregations about the openness of the mouth, in particular the mean, the minimum and the maximum of the mouth openness. The approaches vary in their different time windows with regards to the length of the time window as well as the minimum required events within a time window. The configuration of the described approaches can be seen in Table III – Table X.

The evaluation of the four tested approaches reveals that, for the first configuration, both the Neural Network and Random Forest perform very similar to each other. When the time window is extended for the second approach, the Random Forest can greatly outperform the Neural Network. Both the Random Forest algorithm and the Neural Network can further improve with added aggregated features for the third and fourth configuration. While the fourth configuration raises the minimum required event count and scores the best F1-Score for all the different configurations with a Random Forest approach.

TABLE III. Neural Network Configuration and Evaluation (1)

# of Hidden Nodes	Learning Rate	Iterations	Macro F1 Score
100	0,1	100	0,421671209
100	0,2	100	**0,692872142**
100	0,4	100	0,49980675
100	0,1	20	0,389434236
100	0,2	20	0,404154329
100	0,4	20	0,408630991
100	0,1	160	0,424437618
100	0,2	160	0,481973432
100	0,4	160	0,510131702

TABLE IV. RANDOM FOREST CONFIGURATION AND EVALUATION (1)

# of Trees	Max Depth	Min Leaf Samples	Macro F1 Score
8	32	1	0,469416976
2	32	1	0,609198354
32	32	1	**0,700695199**
32	8	1	0,489023857
8	8	1	0,510131702
2	8	1	0,540695199
2	32	4	0,596979449
8	32	4	0,615180637
32	32	4	0,625025178
2	8	4	0,539904192
8	8	4	0,558837421
32	8	4	0,566706585

TABLE V. NEURAL NETWORK CONFIGURATION AND EVALUATION (2)

# of Hidden Nodes	Learning Rate	Iterations	Macro F1 Score
100	0,1	100	0,42917322
100	0,2	100	0,4430901
100	0,4	100	0,513136
100	0,1	20	0,51614831
100	0,2	20	0,43837494
100	0,4	20	0,4289338
100	0,1	160	0,43087189
100	0,2	160	0,45317711
100	0,4	160	**0,51899583**

TABLE VI. RANDOM FOREST CONFIGURATION AND EVALUATION (2)

# of Trees	Max Depth	Min Leaf Samples	Macro F1 Score
8	32	1	**0,72581615**
2	32	1	0,610908
32	32	1	0,70992889
32	8	1	0,59023321
8	8	1	0,57078645
2	8	1	0,51899583
2	32	4	0,60129555
8	32	4	0,63963819
32	32	4	0,63128689
2	8	4	0,55471677
8	8	4	0,55855759
32	8	4	0,566706585

TABLE VII. NEURAL NETWORK CONFIGURATION AND EVALUATION (3)

# of Hidden Nodes	Learning Rate	Iterations	Macro F1 Score
100	0,1	100	0,42152261
100	0,2	100	0,50116839
100	0,4	100	0,50559143
100	0,1	20	0,44251563
100	0,2	20	0,41031938
100	0,4	20	0,41584999
100	0,1	160	0,50229158
100	0,2	160	0,51476433
100	0,4	160	**0,52495679**

TABLE VIII. RANDOM FOREST CONFIGURATION AND EVALUATION (3)

# of Trees	Max Depth	Min Leaf Samples	Macro F1 Score
8	32	1	**0,74347624**
2	32	1	0,63608707
32	32	1	0,71681175
32	8	1	0,57639503
8	8	1	0,57193689
2	8	1	0,54447684
2	32	4	0,62142392
8	32	4	0,62958668
32	32	4	0,630966
2	8	4	0,53782808
8	8	4	0,55429013
32	8	4	0,56321058

TABLE IX. NEURAL NETWORK CONFIGURATION AND EVALUATION (4)

# of Hidden Nodes	Learning Rate	Iterations	Macro F1 Score
100	0,1	100	0,42231288
100	0,2	100	0,49577651
100	0,4	100	0,49710717
100	0,1	20	0,32834959
100	0,2	20	0,32834959
100	0,4	20	0,41455286
100	0,1	160	0,51472129
100	0,2	160	0,50998433
100	0,4	160	**0,516294**

TABLE X. RANDOM FOREST CONFIGURATION AND EVALUATION (4)

# of Trees	Max Depth	Min Leaf Samples	Macro F1 Score
8	32	1	**0,75287793**
2	32	1	0,63763463
32	32	1	0,73417818
32	8	1	0,58773432
8	8	1	0,57711517
2	8	1	0,55175332
2	32	4	0,62253607
8	32	4	0,64804026
32	32	4	0,64321045
2	8	4	0,55076006
8	8	4	0,57988178
32	8	4	0,57678061

VII. DISCUSSION

In this work we have developed a Cloud-based ML algorithm aiming to detect dangerous driving behaviors. The Microsoft Azure platform was used to test the proposed ML algorithms against each other. Data sets collected within the drive-safely application were used as data basis for developing a threshold-based system for the detection of dangerous driving behaviors.

The development process has been divided into 3 phases. First, it was necessary to understand the existing threshold-based classification procedure. Second, it was important to extract added value from existing data in order to enrich the classification and possibly achieve better results. Whether this is possible, had to be found out in the course of the extensive experiments. Third, the results showed how to handle the data in order to achieve the best possible results.

It can be concluded that the processing of the data is of central importance. Once the data is enriched with information, better results can be achieved. Classification tasks that involve a variety of aggregated features, such as mean values of eye openness that take into account the past of the given time window are performing better than those who do not utilize aggregated features as much. We plan investigate it in further research. To this point a variety of different classification models were tested. Both the configurations of the algorithms themselves as well as the data were mixed up in order to find suitable results for the classification of dangerous states.

The comparison of the two investigated ML approaches revealed that Neural Networks and Random Forests performed very similar to each other with a slight advantage for Random Forests. In addition, it has been recognized that the performance of the classification algorithms improved when additional data were attached to the dataset. In addition, the testing reveals that the sliding window concept for aggregated features improved the performance of the classification when additional aggregations, such as minimum, maximum, and mean values of features are considered. The best performing approach includes aggregated features about the head angles, the mouth openness and the eye openness and is achieved with the Random Forest configuration (Table XI), which has a moderate processing time compared to random forest configurations with 32 individual trees. The corresponding confusion matrix is shown in Fig. 3.

The results are thus based on commodity hardware, in this case smartphones' sensor data. In order to further improve the algorithms or extend the feature set, it would be possible to add further data sources via sensor fusion. There are also recent works on this subject (e.g., [20]) that describe how they combine data from a Mixed Reality (MR) headset (Microsoft HoloLens), a smart-watch, a smartphone, and a data logger that logs data from the vehicle's on-board-diagnostics (OBD) interface. Apart from the fact that it is not allowed in road traffic in many countries, a Mixed Reality headset could provide precise data on head rotations, thereby complementing and making more robust the existing distraction and drowsiness detection based on video data explained in this work. Smart Watches, in turn, can provide data such as the pulse, but also arm movement data to detect distractions where the driver may be looking at the road but is mentally focused on something else, e.g., while changing the radio station, digging around in the pocket looking for something, or typing a text message.

Hence, this would contribute again to distraction detection, but also to drowsiness detection, e.g., as a reduced arm movement, which is typically necessary to keep the vehicle on track, can indicate fatigue [20].

While the data sources mentioned before primarily provide data about the driver and his status, data from the vehicle, on the other hand, can provide information on how the vehicle is used. In [21] the term Quantified Vehicle, which captures sensor data about itself (vehicle usage data) and its environment is introduced. Vehicle usage data could give insights on how risky the vehicle is used, e.g., whether it is often at risk of slipping off the road due to enormous lateral accelerations in curves, or how often and for how long wheels spin. Such events are considered to be part of an aggressive driving style, and this is, as mentioned in Section I, one of the causes of many deaths in road traffic. However, unfortunately such data is currently only available in test vehicles that read the data directly from the vehicle's bus system (e.g., CAN bus).

Services on the market so far mainly use the above mentioned OBD interface, which was actually not developed for this purpose. Theoretically, this interface should provide a lot of interesting data, but in practice with several tested vehicles of different manufacturers, there were only 10-15 relevant signals like vehicle speed, revolutions per minute (RMP), or oil temperature each, which allows driving style analyses, but only to a very limited extent. Recently, however, more and more vehicle manufacturers are discovering their interest in retaining customers with services, which increasingly creates the technical possibilities for third party service developers to gain access to their customers' vehicle data. Even marketplaces have already been created (e.g. caruso-dataplace.com), which offer data from several manufacturers in a uniform data format. This in turn shows that sensor fusion between smartphone sensor data and vehicle data can become a relevant topic, and that there is a clear trend towards turning vehicle and driver data into a business model, as described for example in [22]. In Table XII, we summarize a preliminary list of interesting events we think of extending our solution to in the future.

TABLE XI BEST PERFORMING CONFIGURATION

# of Trees	Max Depth	Min Leaf Samples	Macro F1 Score
8	32	1	0,75287793

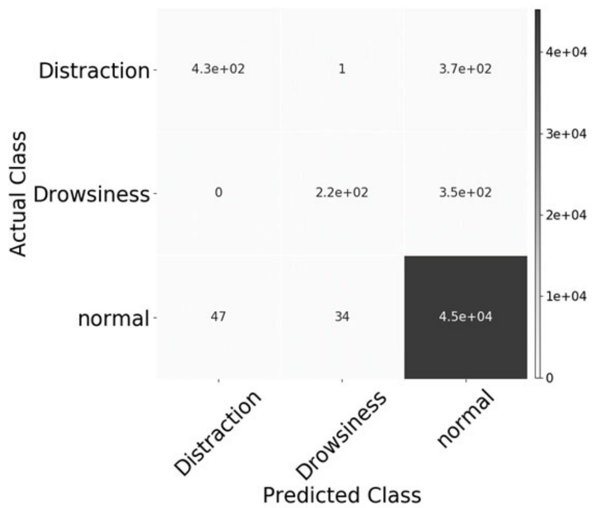

Fig. 3. Confusion matrix for the best performing algorithm

TABLE XII. PARAMETERS FOR THE PROPOSED MODEL EXTENSION

Interesting Events w.r.t risky driving	Data Source
Head movements related to distraction or drowsiness	Smart Glass / Mixed Reality Headset: head rotation sensor
Arm movements that are not intended for driving	Smart Watch: acceleration sensor
Individual arm movement on the steering wheel to keep the vehicle on track, and unusual deviations therefrom	Smart Watch: acceleration sensor
Individual driving style like typical rpm usage and gear change behavior per road segment, to detect unusual behavior	Vehicle OBD interface data (limited set of vehicle sensor data)
Wheel speed per wheel to detect wheel spinning or locking tires	Vehicle CAN bus data (access to more vehicle sensors): wheel speed
Harsh braking, harsh accelerating, harsh cornering	Vehicle CAN bus data (access to more vehicle sensors): vehicle acceleration sensor
Distances to other vehicles and vulnerable road users	Vehicle CAN bus data (access to more vehicle sensors): RADAR/LiDAR/video sensor

VIII. CONCLUSION

The goal of our paper is to present an approach based on ML to extend the existing system with a component that enables learning and adapts to the individual needs of the driver. The long-term vision is to learn from the collected data and make the previously rigid thresholds more flexible. This will be achieved by using ML techniques to establish correlations between the observed behavior and the underlying data and to gain new insights. These findings should be evaluated and can be integrated into the threshold-based system afterwards in order to increase the performance regarding the correct recognition of dangerous situations for the individual driver. In order to meet this requirement, extensive experiments were carried out with the help of the data to determine how an implementation can be done.

The experiments show how important the handling of time series data is with regards to performance and accuracy of the models. Especially when dealing with variable sized time windows problems occur which can skew the resulting classifications. As [15] notes that dynamic time warping, as a representative of a time series analysis methods, can automatically cope with time deformations it seems interesting to test the data analysis step with time series analysis methods. So, we would like to mention that the computation time in the cloud is not critical for the proposed system since it is post processing of the data.

In the future the user confirmed dangerous event information is implemented as additional input besides the captured data by the threshold system. As a starting point it is planned to have questionnaires about the confirmation of dangerous events. That data can afterwards be used to have a more precise classification. Let us assume the cases where the driver denies the proposed dangerous state by the threshold-based system all have certain unknown criteria. These unknown criteria can be linked by the ML component to those cases where the threshold-based system does not perform as well as intended and propose different dangerous states accordingly. This way the ML component of the system can improve on user feedback and give improving recommendations to the threshold-based system regarding its thresholds.

We believe that the research we present is useful for both science and practice to further facilitate the development of smartphone-based vehicle information systems (see e.g., [23] for more details on vehicle IS). Our research indicates that ML technologies such as Neuronal Networks and Random Forests as promising approaches for learning and adapting threshold-based reasoning about individual drivers' states.

ACKNOWLEDGMENT

The research is funded by the Russian Science Foundation (project # 18-71-10065). The architecture model (Section III) has been developed in scope of the Russian State Research #0073-2019-0005. Sections I and VII of this work are partially funded by the InSecTT project (https://www.insectt.eu/). InSecTT has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 876038. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Austria, Sweden, Spain, Italy, France, Portugal, Ireland, Finland, Slovenia, Poland, Netherlands, Turkey. In Austria the project was also funded by the program "ICT of the Future" and the Austrian Federal Ministry for Climate Action, Environment, Energy, Mobility, Innovation and Technology (BMK).

REFERENCES

[1] National Highway Traffic Safety Administration, "Early Estimate of Motor Vehicle Traffic Fatalities in 2019,"

[2] H. O. World, "Global status report on road safety 2018. Geneva: World Health Organization; 2018," 2018.

[3] A. Bolovinou, A. Amditis, F. Bellotti, and M. Tarkiainen, "Driving style recognition for co-operative driving: A survey," in *The Sixth International Conference on Adaptive and Self-Adaptive Systems and Applications*, 2014, pp. 73–78.

[4] N. P. Chandrasiri, K. Nawa, and A. Ishii, "Driving skill classification in curve driving scenes using machine learning," *J. Mod. Transport.*, vol. 24, no. 3, pp. 196–206, 2016.

[5] W. Wang, J. Xi, H. Chen, and R. Yang, "Modeling and Recognizing Driver Behavior Based on Driving Data: A Survey," *Mathematical Problems in Engineering*, vol. 2014, p. 245641, 2014.

[6] A. Hanelt, E. Piccinini, R. W. Gregory, B. Hildebrandt, and L. M. and Kolbe, "Digital Transformation of Primarily Physical Industries - Exploring the Impact of Digital Trends on Business Models of Automobile Manufacturers," in *Wirtschaftsinformatik Proceedings*, 2015. [Online]. Available: https://aisel.aisnet.org/wi2015/88

[7] F. Lindow and A. Kashevnik, "Driver Behavior Identification Based on Smartphone Sensor Data and Machine Learning Methods," *25th Conference of Open Innovations Association (FRUCT)*, 2019, pp. 196–203.

[8] A. Kashevnik, I. Lashkov, A. Ponomarev, N. Teslya, and A. Gurtov, "Cloud-Based Driver Monitoring System Using a Smartphone," *IEEE Sensors Journal*, vol. 20, no. 12, pp. 6701–6715, 2020.

[9] A. Kashevnik, I. Lashkov, and A. Gurtov, "Methodology and Mobile Application for Driver Behavior Analysis and Accident Prevention," *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[10] D. M. Vlachogiannis, E. I. Vlahogianni, and J. Golias, "A Reinforcement Learning Model for Personalized Driving Policies Identification," *International Journal of Transportation Science and Technology*, 2020.

[11] F. K. Adamidis, E. G. Mantouka, and E. I. Vlahogianni, "Effects of controlling aggressive driving behavior on network-wide traffic flow and emissions," *International Journal of Transportation Science and Technology*, 2020.

[12] Y. Ma, Z. Zhang, S. Chen, Y. Yu, and K. Tang, "A comparative study of aggressive driving behavior recognition algorithms based on vehicle motion data," *IEEE Access*, vol. 7, pp. 8028–8038, 2018.

[13] Z. E. A. Elassad, H. Mousannif, H. Al Moatassime, and A. Karkouch, "The application of machine learning techniques for driving behavior analysis: A conceptual framework and a systematic literature review," *Engineering Applications of Artificial Intelligence*, vol. 87, 2020.

[14] C. Naurois, C. Bourdin, A. Stratulat, E. Diaz, and J. Vercher, "Detection and prediction of driver drowsiness using artificial neural network models," *Accident Analysis & Prevention*, vol. 126, pp. 95–104, 2019.

[15] G. Singh, D. Bansal, and S. Sofat, "A smartphone based technique to monitor driving behavior using DTW and crowdsensing," *Pervasive and Mobile Computing*, vol. 40, pp. 56–70, 2017.

[16] D. Yi, J. Su, C. Liu, M. Quddus, and W.-H. Chen, "A machine learning based personalized system for driving state recognition," *Transportation Research Part C: Emerging Technologies*, vol. 105, pp. 241–261, 2019.

[17] J. Yu, Z. Chen, Y. Zhu, Y. Chen, L. Kong, and M. Li, "Fine-Grained Abnormal Driving Behaviors Detection and Identification with Smartphones," *IEEE Trans. on Mobile Comput.*, vol. 16, no. 8, pp. 2198–2212, 2017.

[18] A. Kashevnik, K. Karelskaya, and M. Repp, "Dangerous Situations Determination by Smartphone in Vehicle Cabin: Classification and Algorithms," in *2019 24th Conference of Open Innovations Association (FRUCT)*, 2019, pp. 130–139.

[19] H. Blockeel and J. Vanschoren, “Experiment Databases: Towards an Improved Experimental Methodology in Machine Learning,” in *Lecture notes in computer science, Knowledge Discovery in Databases: PKDD 2007*, J. N. Kok, J. Koronacki, R. Lopez de Mantaras, S. Matwin, D. Mladenič, and A. Skowron, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 6–17.

[20] Lechner, G., Fellmann, M., Festl, A., Kaiser, C., Kalayci, T. E., Spitzer, M., & Stocker, A. "A Lightweight Framework for Multi-device Integration and Multi-sensor Fusion to Explore Driver Distraction." International Conference on Advanced Information Systems Engineering. Springer, Cham, 2019, pp. 80-95.

[21] A. Stocker, C. Kaiser, and M. Fellmann, “Quantified Vehicles,” *Business & information systems engineering*, vol. 59, no. 2, pp. 125–130, 2017.

[22] C. Kaiser, A. Stocker, G. Viscusi, A. Festl, P. Moertl, and M. Glitzner, Eds., ”Quantified cars: An exploration of the position of ICT start-ups vs. car manufacturers towards digital car services and sustainable business models”, 2017.

[23] C. Kaiser, A. Stocker, A. Festl, G. Lechner and M. Fellmann, “A Research Agenda for Vehicle Information Systems”, European Conference on Information Systems, 2018.