Revisiting Gaussian Mixture Models for Driver Identification

Sasan Jafarnejad*, German Castignani† and Thomas Engel*
Interdisciplinary Centre for Security, Reliability & Trust (SnT), Univ. of Luxembourg,
Esch-sur-Alzette, Luxembourg
Email: *{firstname.lastname}@uni.lu, †german.castignani@ext.uni.lu

***Abstract*—The increasing penetration of connected vehicles nowadays has enabled driving data collection at a very large scale. Many telematics applications have been also enabled from the analysis of those datasets and the usage of Machine Learning techniques, including driving behavior analysis, predictive maintenance of vehicles, modeling of vehicle health and vehicle component usage, among others. In particular, being able to identify the individual behind the steering wheel has many application fields. In the insurance or car-rental market, the fact that more than one driver make use of the vehicle generally triggers extra fees for the contract holder. Moreover being able to identify different drivers enables the automation of comfort settings or personalization of advanced driver assistance (ADAS) technologies. In this paper, we propose a driver identification algorithm based on Gaussian Mixture Models (GMM). We show that only using features extracted from the gas pedal position and steering wheel angle signals we are able to achieve near 100% accuracy in scenarios with up to 67 drivers. In comparison to the state-of-the-art, our proposed methodology has lower complexity, superior accuracy and offers scalability to a larger number of drivers.**

***Index Terms*—Driving behavior, driver identification, spectral analysis, Gaussian mixture model**

I. Introduction

The increasing penetration of connected vehicles nowadays has enabled driving data collection at a very large scale. Such a dataset is by definition heterogeneous, including geolocation, speed, inertial motion or even engine information obtained through Car Area Network (CAN). Many telematics applications have been also enabled from the analysis of those datasets and the usage of Machine Learning techniques, including driving behavior analysis, predictive maintenance of vehicles, modeling of vehicle health and vehicle component usage, among others. In particular, being able to identify the individual behind the steering wheel appears as valid research problem to solve many application fields. In the insurance or car-rental market, the fact that more than one driver make use of the vehicle generally triggers extra fees for the contract holder. In such a case, the goal is not to deanonymize the driver but to predict, with a certain probability, the number of different users making use of the vehicle. Another use case scenario arises in modern vehicles, including future autonomous vehicles, where the number of comfort-related features has increased substantially. Being able to identify different drivers may enable the automation of comfort settings (e.g., seats and mirrors position, air-conditioning, adaptive driving-assistance system profiles) In this paper, we propose a driver identification algorithm based on Gaussian Mixture Models (GMM). Compared to state-of-the-art research on driver identification, we prove that the proposed mechanism has a lower computational cost allowing at the same time scalability. Moreover, since only gas pedal position and steering wheel angle signals are used without the need of extra location or speed-based features, privacy can be preserved.

The remainder of the paper is organized as follows: Section II presents the related work, Section III gives an overview of the data-set used in our evaluations. In Section IV we describe our proposed methodology and evaluation strategy. In Section V we present the evaluation results. Lastly, in Section VI we conclude and present the future directions.

II. Related Work

The first use of GMM for driver identification is attributed to Wakita et al. [2]. They identify drivers by using behavioral signals collected while driver performs the ”car following” task. They explore two approaches a) physical driving models, b) modeling based on the distributions of driving signals using GMM. They observe that in this task GMM outperforms physical models. In a follow-up work Miyajima et al. [3] propose to use Cepstral based features for driver identification. Cepstral based features, especially Mel-frequency cepstral coefficients (MFCCs) are well studied for speech and speaker recognition [11]. In their experiments, they discover that using Cepstral coefficients from gas and brake pedal is well suited for driver identification. A very similar approach is taken in [5], however, they evaluate their methodology on a different data-set. These experiments prove the efficacy of GMM with Cepstral features. Although these works show in general good performance of GMM in driver identification, the data they use for their experiments comes from pressure sensors that are retrofitted into highly instrumented vehicles which are not present in vehicles on the market. In the work we propose in this paper, we make use of data coming from CAN-bus through Onboard diagnostic (OBD) port, which by law is mandatory in vehicles manufactured since 1995.

Meng et al. [1] propose to use dynamical models to represent driving behavior for driver identification purposes. They take three signals of Steering Wheel Angle (SW), gas and

TABLE I
An overview of driver identification literature

Reference	Data-set	Signals	Features	Model	Result
Meng et al. [1]	Simulator	Steering, acceleration, braking	FFT	HMM	75% for 7 drivers
Wakita et al. [2]	Simulation CIAIR	Gas/brake pedals headway distance	Raw	Helly, OV GMM	81% for 12 drivers simulator 73% for 30 drivers real car
Miyajima 2007 [3]	Simulator CIAIR	Gas/brake pedals headway distance velocity	Raw Cepstral	Optimal Velocity (OV) Gaussian Mixture Model (GMM)	89.6% for simulator 76.8% for 276 drivers
Qian et al. [4]	Simulator	Gas/brake pedals, steering	FFT, PCA ICA	SVM	85% for 7 drivers
Özturk et al. [5]	UYANIK	Gas/brake pedals headway distance	Cepstral	GMM	85.21% for 3 drivers
Zhang et al [6]	Simulator	Gas pedal, steering	Raw	HMM	85% for 20 drivers
Del campo et al. [7]	UYANIK	Gas/brake pedal	Cepstral	MLP	84.6% for 3 drivers
Martínez et al [8]	UYANIK	12 based on CAN-bus, 6 based on IMU, headway distance	Cepstral Spectral etc.	ELM	96.95% for 3 drivers 84.36% for 11 drivers
Enev et al. [9]	Collected by UCSD	14 signals from CAN-bus Powertrain,Dynamics, Pedals,Steering	Statistical, FFT, etc.	SVM, Random Forest Naive Bayes, KNN	100% for 15 drivers
Jafarnejad et al. [10]	UYANIK	5 signals from CAN-bus	Statistical, Cepstral	SVM, Random Forest AdaBoost, Extra Trees	95% for 5 drivers 89% for 15 drivers 82% for 35 drivers

brake pedals, and perform Fast Fourier Transform (FFT) on each signal and use that as the feature vector to train a Hidden Markov Model Hidden Markov Model (HMM) for each driver. In a more recent study, Zhang et al. [6] also use HMM for driver identification purposes. The downside of using HMM-based methods is the high computational needs and complexity of the algorithms. Qian et al. [4] compare FFT, Independent Component Analysis (ICA) and Principal Component Analysis (PCA) for preprocessing and feature extraction, and propose to use Support Vector Machine (SVM) for driver identifications. They identify that FFT is more suitable than the alternatives, and achieve an accuracy of about 85% for 7 drivers.

Del Campo et al. use Multi-Layer Perceptron (MLP) and focus on real-time driver identification, for which they also develop a specific hardware implementation. In another work, the authors [12] perform systematic feature selection and employ Extreme Learning Machine Extreme Learning Machine (ELM) for driver identification.

Enev et al. perform an extensive feature analysis of driver identification and use various ensemble methods such as Random Forest (RF) and an extensive set of features they manage to achieve a very good identification accuracy of 100%. Since their aim is to demonstrate the privacy issues caused by sharing driving data, they have used every computationally expensive methodology which is not suitable for practical purposes. As an example, in order to classify n drivers they require fitting n^2 models. In our previous work [10], we propose a methodology for driver identification based on AdaBoost. We use 5 signals available from CAN-bus, including Cepstral features from the gas pedal and steering wheel. Although that methodology provides very high accuracy (i.e., 95, 89, 82 percent accuracy for 5, 15, 35 driver respectively), it suffered from two drawbacks that we greatly improved in this work, a) the need to retrain the models after addition of each new driver, b) the need for large amounts of test-data to achieve a good level of accuracy.

Table I presents an overview in terms of methodological approach and obtained performance of works mentioned above.

III. Data-set

In order to validate the methodology developed in this work, we use the UYANIK data-set, which has been collected under the shared framework of Drive-Safe Consortium (Turkey) and NEDO (Japan) International Collaborative Research [13] [14]. The data collection was performed using a heavily instrumented Renault Megane. The data includes video and audio, location data, CAN-bus data, Inertial Measurement Unit (IMU) data, laser range-finder and pressure sensors underneath gas and brake pedals. Up to 105 participants drove a route which consisted of 25 km stretch including city and motorway, lasting about 45 minutes on average. During the experiment, participants had to perform various secondary tasks which resemble occasional distractions during a typical daily driving experience (For more details refer to [13], [15]). Since this work intends to propose a solution that can be widely deployed with the lowest cost impact, therefore we will not use any of the signals coming from retrofitted sensors and only rely on the CAN-bus data, universally available in vehicles in the market. Moreover, knowing that various vehicles provide access to different set of sensors from CAN-bus we limit ourselves to only the most important signals, namely those directly operated by the driver, steering wheel and pedals. More precisely from

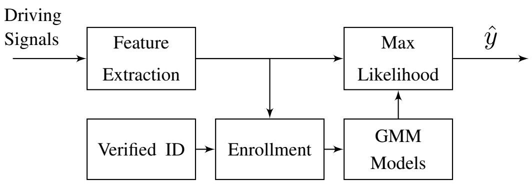


Fig. 1. Driver Identification System

the UYANIK data-set we only use SW and Percentage Gas Pedal (GP) signals, these signals are sampled 32 times per second (32Hz). We also performed a pre-processing step to ensure that driving sessions containing a high proportion of corrupt data are discarded. Then in order to have a balanced dataset, we discard recordings that are too short or long. In the end, we are left with 67 different drivers containing a fully valid data-set to perform our experiments.

IV. Methodology

In this section, we describe the proposed methodology which phases are depicted in Figure 1. First we explain the GMM for driver identification, and model fusion mechanism then we go over the feature extraction and analysis, and in the end the evaluation criteria and strategy.

A. The Gaussian Mixture Driver Model

We define the goal of driver identification as the assignment of a driving trace $\boldsymbol{X}$ to its corresponding driver y. We also assume that the set of target drivers is finite and that we have information about all candidate drivers. We propose to use GMMs for driver identification as they are well studied in the literature, in particular for speaker recognition [11].

A GMM is a weighted sum of M component densities (Equation 1).

$$p\left(\boldsymbol{x}|\lambda\right) = \sum_{i=1}^{M} \phi_i \mathcal{N}_i\left(\boldsymbol{x}\right) \tag{1}$$

where $\boldsymbol{x}$ is a D-dimensional feature vector, $\mathcal{N}_i(\boldsymbol{x}), i = 1, \cdots, M$, are the component densities and $\phi_i, i = 1, \cdots, M$, are the mixture weights. Each component density is a D-variate Gaussian of the form (Equation 2):

$$\mathcal{N}_i(\boldsymbol{x}) = \frac{1}{(2\pi)^{D/2}|\Sigma_i|^{1/2}} \exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)'\Sigma_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i)\right\} \tag{2}$$

where $\boldsymbol{\mu}_i$ is the mean vector and $\boldsymbol{\Sigma}_i$ is the covariance matrix. The mixture weights should satisfy the constraint $\sum_{i=1}^{M} p_i = 1$.

$$\lambda = \{\phi_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\} \quad i = 1, \cdots, M. \tag{3}$$

Therefore each driver is represented by a GMM and is referred to by their model λ.

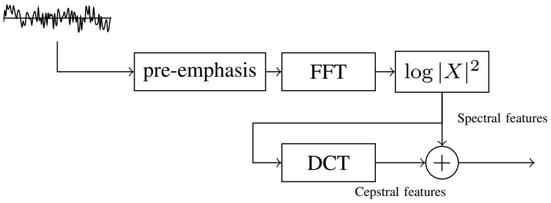


Fig. 2. Feature Extraction

B. Driver Identification

For driver identification, a group of S drivers is represented by S GMMs, $\lambda_1, \lambda_2, \cdots, \lambda_S$. Here our goal is to find the model which has the maximum a posteriori probability for a given set of observations.

$$\hat{y} = \underset{1\le k\le S}{\arg\max}\, P_r(\lambda_k|\boldsymbol{X}) = \underset{1\le k\le S}{\arg\max}\, \frac{p(\boldsymbol{X}|\lambda_k)P_r(\lambda_k)}{p(\boldsymbol{X})} \tag{4}$$

Where $\boldsymbol{X}$ is vector of observations at prediction time ($\boldsymbol{X} = \{\boldsymbol{x}_t\}_{i=1}^{T}$). Assuming that all the candidate drivers are equally likely to be the actual driver ($P_r(\lambda_k) = 1/S$) and taking into consideration that $p(\boldsymbol{X})$ is the same for all driver models, Equation 4 reduces to:

$$\hat{y} = \underset{1\le k\le S}{\arg\max}\, p(\boldsymbol{X}|\lambda_k) \tag{5}$$

Assuming independence between measurements and using log to facilitate computations we have:

$$\hat{y} = \underset{1\le k\le S}{\arg\max} \sum_{t=1}^{T} \log p(\boldsymbol{x}_t|\lambda_k) \tag{6}$$

where $p(\boldsymbol{x}_t|\lambda_k)$ is defined by Equation 1.

C. Model Fusion

We consider two signals, gas pedal (GP) and steering wheel (SW). We chose these signals because they are: 1) directly operated by driver 2) available through CAN-bus. Since these two signals come from two controllers that are operated separately, we model each with a GMM. We use the notation λ_G to refer to gas pedal operation model and λS for steering operation model. Moreover, because the two resulting models (λ_G and λ_S) are not equally informative, we need to give a higher weight to the model that is a better predictor of the driver. We use a parameter called α to indicate the weight that we associate to each model's likelihoods. Then at identification time, we combine their log-likelihoods linearly, the ratio is controlled by the parameter α. In this case, the Equation 6 becomes as below:

$$\hat{y} = \underset{1\le k\le S}{\arg\max}\{\alpha \log p(\boldsymbol{X}_G|\lambda_{G,k}) + (1-\alpha)\log p(\boldsymbol{X}_S|\lambda_{S,k})\} \tag{7}$$

where $\boldsymbol{X}_G, \boldsymbol{X}_S$ refer to gas pedal operation feature vectors and steering operation feature vectors respectively, and

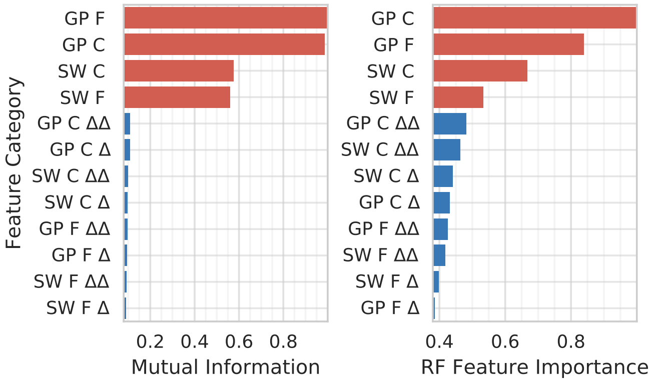


Fig. 3. Feature importance

$\lambda_{G,k}, \lambda_{S,k}$ refer to the kth driver's gas pedal and steering models. The idea is that using the right weight will improve the accuracy. We propose to find the optimal α empirically in Section V.

D. Feature Extraction

A sliding window approach is used for feature extraction. We use a 2 seconds long Hamming-window and stride of 1 second, therefore two consecutive window frames have 50% overlap. As it is shown in Figure 2 signal from each window frame is pre-emphasized[1] to enhance higher frequency components, we compute log-magnitude-spectrum of the signal. From here we perform two different set of operations to obtain two groups of features, spectral features and cepstral features. For spectral features we consider log-power-magnitude of the signal for each 1Hz range to be a feature, up to 8Hz however, because higher frequencies have smaller magnitude we compress them in larger ranges in this case, 8-12Hz and, 12Hz and above. We also include the sum of all energies as a separate feature, therefore we would have 11 spectral features. To obtain cepstral features we apply Discrete Cosine Transform (DCT) to log-power-magnitude of signal and mean normalize it. In this work, we only keep the first 8 cepstral coefficients as features. Like spectral features, we also consider the sum of energy among the cepstral coefficients to be an extra feature as well. This will leave us with 9 cepstral features. This results in a feature vector of dimension 20 per signal. We also scale features using the following formula:

$$X_{scaled} = \frac{X_{raw} - \bar{X}_{raw}}{std(X_{raw}) + \epsilon} \tag{8}$$

to have each feature with a mean of close to zero and variance of one. ϵ is a very small value (10^{-7}) added to avoid division by zero. In our experiments, we scale the data only based on the training data, and apply the same scale to the test data.

E. Feature Analysis

In initial experiments, we considered adding Δ and $\Delta\Delta$ of both cepstral and spectral features. Delta cepstral features are well studied in speech recognition field. They have shown to improve accuracy by adding dynamic information to cepstral coefficients which in turn helps explain temporal dependency between the frames [16], [17]. Addition of Δ and $\Delta\Delta$ will increase the number of features to 60 per signal. Here we perform analysis to quantify and validate positive contribution of each of our signals and feature categories. We use two measures, mutual information and random forest feature importance, for both of which implementations from scikit-learn were used [18]. Results are presented in Figure 3. We can observe that both signals are important however, it is clear that GP presents higher importance. When it comes to comparing feature categories, cepstral features show a slight advantage however the gap is not significant. It is also clear that δ features do not play an important role and have low importance. We also validated this in our preliminary results as we would usually get better results without δ features. As a result we decided to remove them altogether for the sake of simplicity and performance.

F. Evaluation Method

There are two important points we take into account in evaluating our method. Firstly, since we perform identification in small sets of drivers c (e.g. 5, 10, 35), it is important to account for unwanted side effects. For example, it could be that one set of 5 drivers have very distinct driving styles and therefore be easily discriminated, while another set of 5 drivers have similar driving styles and be difficult to discriminate. To prevent biasing our results with such phenomena we repeat each experiment with 30 random samples of k drivers. For each evaluation in order to use the whole data-set for both training and testing, we employ a cross-validation approach. In particular hold-one-out cross-validation, in which we first segment each trip into 10 slices. At each fold, we hold-out one slice from each of selected drivers, train the models, and test on the held-out slice. We use accuracy as the evaluation metric, as defined below:

$$acc = \frac{1}{k}\sum_{i=1}^{k} \mathbb{1}(y_i = \hat{y}_i) \tag{9}$$

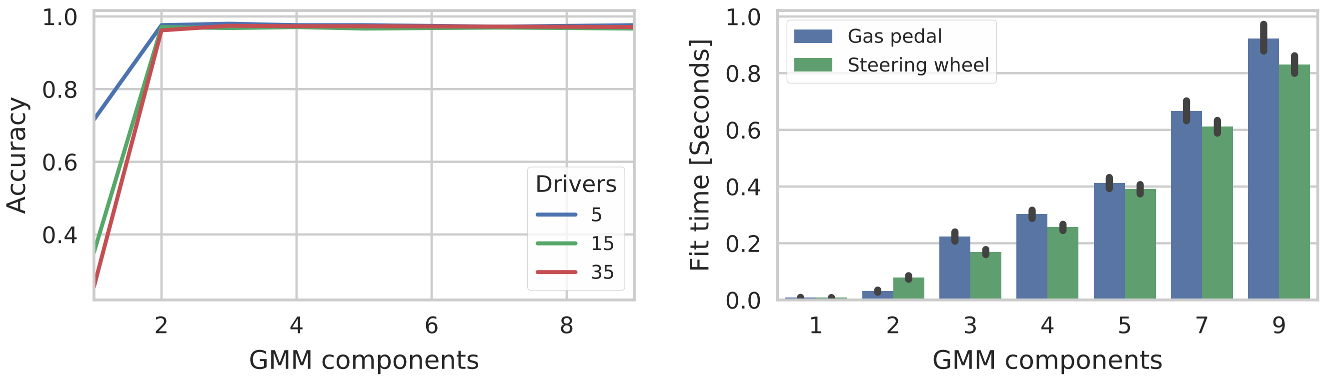
where $\mathbb{1}$ is the indicator function, $\hat{y}_i$ and y_i are respectively predictions and true driver for i^{th} driving trace.
The overall accuracy score for an experiment is the average of all cases as follows:

$$acc_{score} = \frac{1}{R \cdot L}\sum_{r=1}^{R}\sum_{l=1}^{L} acc_{r,l} \tag{10}$$

where $L = 10$ is number of cross-validation folds and $R = 30$ is number of repetitions. In the rest of paper, we refer to this score as accuracy.

In our experiments we use notions of *training window* and *decision window*, the former refers to the amount of data used for training and the latter refers to the amount of data used for testing. Earlier we mentioned that we slice each driver's trip into 10 slices, and at each cross-validation fold 9 slices are

[1] $y[n] = x[n] - \beta x[n-1]$



(a) Effect of variations in number of GMM components

(b) Average fit-time for each model λ

Fig. 4. Model Parameters

considered to be part of training-set and 1 slice part of the test-set. In order to perform various analysis sometimes we truncate the train-set and test-set, for example, training window of 10 minutes and decision window of 1 minute means that from the training-set we use the first 10 minutes of the trip for fitting the models and from the test-set we use the first 60 seconds for predicting the driver.

V. Results

In this section, we will evaluate our proposed methodology from various aspects and analyze how model parameters affect the performance.

A. Model Parameters

We seek optimal values for two parameters M which indicated the number of GMM components and α which is the ratio by which we linearly combine the likelihoods from steering and gas pedal models. First, to determine the optimal number of GMM components we perform a set of experiments by varying number of GMM components. We consider driver models with 1 to 9 Gaussian components and fixed $\alpha = 0.5$, and evaluate their performance. From the results presented in Figure 4a we can see that accuracy plateaus starting from 2 GMM components and the best result is obtained with 3 GMM components.

Depending on the application some time constraints maybe need to be satisfied by the driver identification solution (e.g., detecting fraud, personalizing car configurations per driver). Therefore we also investigate the training time of the models. Figure 4b shows the obtained results. As one would expect the fitting time increases with the number of GMM components. So for each driver total training time would be the sum of gas pedal (λ_G) and steering (λ_G) models. Similarly, for a scenario with 5 drivers, the total training time is going to be approximately 5 times the fit time of a single driver. Since with $M = 3$ the required computational time to train the models is still reasonable in the following sections we always use models with 3 GMM components.

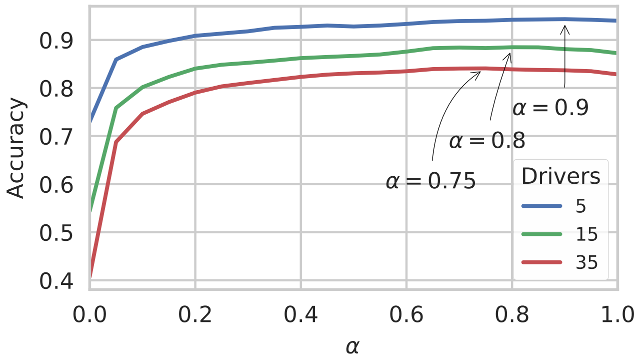


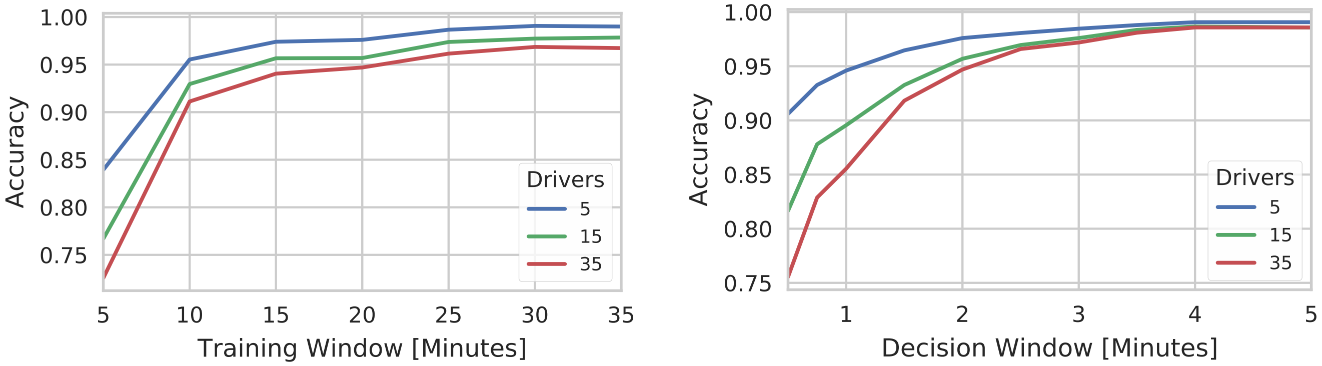
Fig. 5. How α affects accuracy

Second, to obtain the optimal combination of gas pedal and steering features, we vary α from 0 to 1 and evaluate our method. Having $\alpha = 0$ is equivalent to only using λ_S and $\alpha = 1$ would be equivalent to just using λ_G. The corresponding results are presented in Figure 5. As it can be observed, the way α affects the results varies with the number of drivers. It is clear that as the number of drivers goes up λ_S has a more positive effect on the results, this is in line with the findings in our previous work [10].

Additionally, knowing that drivers operate gas pedal and the steering wheel only at parts of their drive, for example when the road is straight no steering is required, and there are times that driver lifts their feet from the pedal either to slow down or to keep a constant speed; At such moments data from the corresponding signals is not quite informative. Therefore we also considered dynamically switching between the models (λ_G and λ_S) based on activity in signals (average energy) however we did not observe significant improvements in performance.

B. Sensitivity Analysis of Training Window

One of the important factors in driver identification is driver enrollment. In order to add a new driver to the system, we



(a) How training window affects accuracy

(b) How decision window affects accuracy

Fig. 6. Identification Performance

need to collect sufficient amount of data from the driver to fit a model. To find out how much data is enough we perform experiments by varying the amount of train-data in 5-minute steps from 5 to 35 minutes of driving data and measure the performance with a fixed 2 minutes of test-data. The results are presented in Figure 6a. As one can see 5 minutes of training data yields poor results, however starting from 10 minutes and above, performance crosses 90%. From this plot, we can conclude that with just 10 minutes of driving data we can obtain good accuracy and as we collect more data from the driver it is possible to improve the model [19].

C. Sensitivity Analysis of Decision Window

Driving is a repetitive task, however, it takes a few minutes until one repeats various maneuvers during a driving session. For example, consider driving on a straight stretch of road, there is no steering maneuvers and only a few pedal operations. Therefore to obtain more reliable results, a larger decision window is required. Moreover, in applications such as theft detection and comfort, you would want to be able to accurately identify the driver in the least time possible. Therefore we evaluate the impact of test-data on performance. To do so we run experiments with a fixed training data of 20 minutes but with variable sizes of decision window. We cover variations from 30 seconds to 5 minutes. A 60 seconds long decision window leads to good results but starting from 90 seconds for all scenarios we obtain more than 90% accuracy.

TABLE II
SUMMARY OF RESULTS

Train	Decision window	Drivers				
Minutes	Minutes	5	15	35	50	67
15	1	0.943	0.880	0.836	0.825	0.819
	2	0.974	0.956	0.940	0.932	0.934
	4	0.990	0.987	0.984	0.983	0.983
30	1	0.967	0.932	0.897	0.887	0.877
	2	0.990	0.977	0.968	0.963	0.959
	4	0.999	0.997	0.996	0.995	0.995

D. Final Results

Here we present a summary of the results obtained in our experiments. The results are presented in Table II. We selected two cases, one with 15 minutes of training data to fit the models representing a use case that not much of data from drivers is available and one with 30 minutes of training data, representing a case with higher data availability. For each case then we present accuracies for three decision window lengths, 1, 2 and 4 minutes. The choices are made to cover a wide range of applications and accuracies we can get with our solutions under these varying conditions. We can see that even with 15 minutes of training data, after two minutes we can reach accuracies of over 95% for 5 and 15 drivers and 94% for 35 drivers. We also include extended experiments with 50 and 67 (the whole data-set). We can observe that with further increase in number of drivers accuracy slightly decreases, however, this to large degree can be compensated with collection of more data both for training and for prediction.

E. Comparison with the previous work

In comparison to our previous work, we significantly improved the accuracy. In particular the accuracy increase by 5.15, 12.02, 21.46 percent for 5, 15, 35 drivers respectively. Moreover, we reduced the number of signals used down to 2 and 20 features per signal. Another major improvement over the previous method is the fact that addition of new drivers to the system is as easy of collecting training data and fitting two GMMs for it. In our previous work we needed to retrain every time we need to add a new driver to the system.
This method is also more privacy-preserving because only using the speed and knowing workplace or residential address of a person one could infer their destination using only the speed [20]. Therefore with this method, we cannot learn anything more than pedal or steering wheel operation patterns, which do not reveal much information about person's lifestyle or identity.

VI. Conclusion

In this work, we proposed a methodology for driver identification that delivers state-of-the-art accuracy while being computationally light-weight. We only use two signals (gas pedal position and steering wheel angle) that are easily accessible from CAN-bus. Inspired by the speech recognition research we extracted cepstral and spectral features using a sliding window mechanism. The resulting feature vectors were used to fit two GMM for each driver (one per signal). At identification time, feature vectors are extracted from the driving data and the driver is predicted based on the maximum likelihood estimation principles. We linearly combine likelihoods from two models, for which we obtain the optimum ratios (α) empirically. We provided analysis covering variations in model parameters, training and prediction conditions. We showed that with 30 minutes for driving data for training and 4 minutes of driving data for prediction our proposed method achieves an accuracy of over 99% for scenarios with 5, 15 and 35 drivers. For future work we propose investigations on portability of the GMM models between the vehicles, unfortunately in our data-set every one drives the same car however, it would be of interest to see if our proposed method will also work if the driver changes the cars, this would be of interest to logistic companies when driver is not tied to a single vehicle or when an insurance customer changes their car ideally one would want to be able to use the same model across various vehicles.

Acknowledgment

The authors would like to thank Dr. Hüseyn Abut and VPALAB of Sabanci University for providing the dataset.

References

[1] X. Meng, K. K. Lee, and Y. Xu, "Human Driving Behavior Recognition Based on Hidden Markov Models," *2006 IEEE Int. Conf. Robot. Biomimetics*, pp. 274–279, 2006.

[2] T. Wakita, K. Ozawa, C. Miyajima, K. Igarashi, K. Itou, K. Takeda, and F. Itakura, "Driver identification using driving behavior signals," *IEICE Trans. Inf. Syst.*, vol. E89-D, no. 3, pp. 1188–1194, mar 2006.

[3] C. Miyajima, Y. Nishiwaki, K. Ozawa, T. Wakita, K. Itou, K. Takeda, and F. Itakura, "Driver modeling based on driving behavior and its evaluation in driver identification," *Proc. IEEE*, vol. 95, no. 2, pp. 427–437, feb 2007.

[4] H. Qian, Y. Ou, X. Wu, X. Meng, and Y. Xu, "Support Vector Machine for Behavior-Based Driver Identification System," *J. Robot.*, vol. 2010, pp. 1–11, 2010.

[5] E. Öztürk and E. Erzin, "Driver Status Identification from Driving Behavior Signals," in *Digit. Signal Process. In-Vehicle Syst. Saf.*, J. H. L. Hansen, P. Boyraz, K. Takeda, and H. Abut, Eds. New York, NY: Springer New York, 2012, pp. 31–55.

[6] X. Zhang, X. Zhao, and J. Rong, "A study of individual characteristics of driving behavior based on hidden markov model," *Sensors & Transducers*, vol. 167, no. 3, pp. 194–202, 2014.

[7] I. Del Campo, R. Finker, M. V. Martinez, J. Echanobe, and F. Doctor, "A real-time driver identification system based on artificial neural networks and cepstral analysis," in *Proc. Int. Jt. Conf. Neural Networks*. IEEE, jul 2014, pp. 1848–1855.

[8] M. V. Martínez, J. Echanobe, I. Campo, M. Martinez, J. Echanobe, and I. del Campo, "Driver Identification and Impostor Detection based on Driving Behavior Signals *," *2016 IEEE 19th Int. Conf. Intell. Transp. Syst.*, pp. 372–378, nov 2016.

[9] M. Enev, A. Takakuwa, K. Koscher, and T. Kohno, "Automobile Driver Fingerprinting," *Proc. Priv. Enhancing Technol.*, vol. 2016, no. 1, jan 2016.

[10] S. Jafarnejad, G. Castignani, and T. Engel, "Towards a Real-Time Driver Identification Mechanism Based on Driving Sensing Data," *20th Int. Conf. Intell. Transp. Syst.*, no. October, p. 7, 2017.

[11] D. A. Reynolds, "Speaker identification and verification using Gaussian mixture speaker models," *Speech Commun.*, vol. 17, no. 1-2, pp. 91–108, aug 1995.

[12] M. V. Martinez, I. D. Campo, J. Echanobe, and K. Basterretxea, "Driving Behavior Signals and Machine Learning: A Personalized Driver Assistance System," in *IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC*, vol. 2015-October, sep 2015, pp. 2933–2940.

[13] H. Abut, H. Erdoğan, A. Erçil, A. B. Çürüklü, H. C. Koman, F. Tas, A. Ö. Argunşah, B. Akan, H. Karabalkan, E. Çökelek *et al.*, "Data collection with" uyanik": too much pain; but gains are coming," 2007.

[14] C. Miyajima, T. Kusakawa, T. Nishino, N. Kitaoka, K. Itou, and K. Takeda, "On-going data collection of driving behavior signals," in *In-Vehicle Corpus and Signal Processing for Driver Behavior*. Springer, 2009, pp. 45–54.

[15] S. Jafarnejad, G. Castignani, and T. Engel, "Non-intrusive distracted driving detection based on driving sensing data," in *Proceedings of the 4th International Conference on Vehicle Technology and Intelligent Transport Systems - Volume 1: VEHITS,*, INSTICC. SciTePress, 2018, pp. 178–186.

[16] S. Furui, "Speaker-independent isolated word recognition based on emphasized spectral dynamics," *ICASSP '86. IEEE Int. Conf. Acoust. Speech, Signal Process.*, vol. 11, pp. 1991–1994, 1986.

[17] B. Hanson and T. Applebaum, "Robust speaker-independent word recognition using static, dynamic and acceleration features: experiments with Lombard and noisy speech," *Int. Conf. Acoust. Speech, Signal Process.*, pp. 857–860, 1990.

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[19] M. Song and H. Wang, "Highly efficient incremental estimation of gaussian mixture models for online data stream clustering," in *Intelligent Computing: Theory and Applications III*, vol. 5803. International Society for Optics and Photonics, 2005, pp. 174–184.

[20] B. Firner, S. Sugrim, Y. Yang, and J. Lindqvist, "Elastic Pathing: Your Speed is Enough to Track You."