2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)
Windsor Oceanico Hotel, Rio de Janeiro, Brazil, November 1-4, 2016

Driver Identification and Impostor Detection based on Driving Behavior Signals*

M.V. Martínez, J. Echanobe, *Member, IEEE*, and I. del Campo, *Member, IEEE*

Abstract— **The progressive integration of Advanced Driver Assistant Systems (ADAS) into vehicles has contributed significantly to increasing safety and comfort levels of the driver. The need to adapt the vehicle to the preferences and requirements of the driver leads to the development of individualized ADAS. Automatic identification of the driver is a key factor in the design of these systems. In this work, a driver identification model with impostor detection capability is proposed. This approach is based on non-intrusive information from driving behavior signals, and an extreme learning machine (ELM) network. The performance of the system is evaluated on the basis of groups of different number of known drivers, and possible impostor drivers. Identification rates are greater than 80% for every group category tested, and still above 90% for groups of two and three drivers. The impostor detection rate is above 80% when the car has a single genuine driver. This rate decays in inverse proportion to the number of authorized drivers, but it is greater than 50% in all cases.**

I. Introduction

In the context of Intelligent Mobility, the so-called Advanced Driver Assistant Systems (ADAS) are the technological steps towards high-level automated and connected driving. In the last three decades, the progressive integration of ADAS into vehicles has contributed significantly to increasing safety and comfort levels of the driver. In this respect, the individualization of these systems is an important research aspect to adapt assistance functions to the individual preferences and requirements, or special needs of the driver.

The capability of adapting vehicles to their owners requires a primary kernel module to identify the driver. There are several products on the market related to driver identification. For example, [1] commercializes an RFID-based device for insurance and fleet management that monitors vehicle location and driving behavior. Collected data, along with driver identity, are sent to a server for real-time monitoring. The main disadvantage of this system is the risk of attempts at impersonation of RFID technology. There are some driver identification approaches based on biometrics, as face scan [2], voice processing [3], audio and video data fusion [4], or sitting posture [5]. However, all of them require additional pieces of equipment to be installed inside the car. Moreover, some drivers consider that video and audio recordings are intrusive. To overcome this problem, a number of research projects consider only non-intrusive information, i.e., driving behavior signals, mainly CAN bus signals and sensor recordings, to build models of driving style. Some of these solutions, as those using Gaussian mixture models (GMM) [6] or support vector machines (SVM) [7], achieve good performance. However, the algorithms they are based on are too complex and computationally expensive for in-vehicle embedded systems, with strong constraints in size and power consumption. A recent driver identification model with interesting results has been formulated as an Extreme Learning Machine (ELM), a particular neural network trained with a very fast single iteration algorithm [8]. The solutions presented in this paper are built from a driver identification scheme of this type, optimized in terms of the number of variables and the size of the network.

Considering security, in addition to the driver identification function, the kernel of an individualized assistance system should be provided with an impostor detection module. The purpose of this module is the verification of authorized drivers, in scenarios like a family car or a company fleet. In practice, the kernel should not be a mere classification component operating within a closed set of drivers. Instead, it should have an open set recognition purpose, so that it was able to identify some driving styles in a larger space of drivers. Therefore, the kernel will be a multi-class classifier with reject option, and its performance will be described by a tradeoff between the error rate and the reject rate [9]. In this work, an ELM scheme for driver identification with impostor detection capability is proposed. The system uses driving behavior signals, i.e., non-intrusive information coming from a large amount of resources nowadays available in commercial vehicles.

The paper is organized as follows. Section II describes the extreme learning machine architecture and algorithm, as well as the data collection, the set of preselected driving signals, and the collection of informative features. In Section III, the design and performance of the closed-set driver identification system, optimized in terms of the size of the ELM network, is presented. Section IV discusses the proposed ELM approach for impostor detection, and presents the experiments and the results. Finally the main conclusions of the work are explained in Section V.

II. ELM Architecture and Driving Variables

The impostor detection system proposed in this work is built starting from the design of a driver identification model previously conceived [8]. This is a closed-set classifier based on driving behavior signals whose objective is to identify the driver from among a given group of known drivers. The model consists in a neural network trained with the Extreme Learning Machine algorithm.

We present here the main aspects of this scheme: the learning algorithm and the ELM architecture, and the data collection and the driving signals and features.

A. Extreme Learning Machine

An Extreme Learning Machine consists of a single-hidden-layer feedforward neural network. The weights and

* This work has been partially funded by the Basque Government under Grant IT733-13, and the Spanish Ministry of Economy and Competitiveness under Grant TEC2013-42286-R.

biases that link the inputs with the neurons in the hidden layer are random numbers, so they are independent of the target application. The algorithm is a very direct and fast learning procedure because the hidden layer does not need to be tuned. The weights linking the hidden layer with the output layer are computed by solving a linear equation system.

Let us consider a neural network with *n* inputs, *m* outputs, and *L* nodes in the hidden layer, such as the network depicted in Fig. 1. The output for generalized ELM is [10]

$$y(\mathbf{x}) = \sum_{i=1}^{L} \beta_i h_i(\mathbf{x}) = \mathbf{h}(\mathbf{x})\boldsymbol{\beta} \tag{1}$$

Without loss of generalization, a single output node ($m = 1$) is taken in (1). The vector of weights, $\boldsymbol{\beta} = [\beta_1,\ldots, \beta_L]^T$, links the hidden nodes with the output node, and $\mathbf{h}(\mathbf{x}) = [h_1(\mathbf{x}),\ldots, h_L(\mathbf{x})]$ is the output vector of the hidden layer for a given input $\mathbf{x} \in \mathbb{R}^n$.

The output of the *i*th hidden node is

$$h_i(\mathbf{x}) = f(\mathbf{a}_i, b_i, \mathbf{x}) = s(\mathbf{a}_i \mathbf{x} + b_i), \tag{2}$$

with s($\mathbf{a}_i$, b_i, $\mathbf{x}$) being the sigmoid activation function, $\mathbf{a}_i \in \mathbb{R}^n$ the random weight vector connecting the inputs $\mathbf{x} = [x_1,\ldots, x_n]$ with the *i*th hidden node, and $b_i \in \mathbb{R}$ the random bias of the *i*th hidden node.

The main difference between ELM and traditional learning approaches is that the hidden layer need not be tuned; its weights and bias are randomized generated values that do not change during the training process.

Learning with ELM is a straightforward procedure that aims at computing the vector of output weights, $\boldsymbol{\beta}$ in (1), for each output node. Given a set of *K* training samples, ($\mathbf{x}_j$, $\mathbf{t}_j$), $1 \le j \le K$, where $\mathbf{x}_j$ is the *j*th input vector, and $\mathbf{t}_j$ is the corresponding target vector, learning is performed by solving (1) for the set of training samples

$$\mathbf{T} = \mathbf{H}(\mathbf{x})\mathbf{B} \tag{3}$$

with $\mathbf{H}$ being the hidden layer output matrix

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}(\mathbf{x}_1) \\ \vdots \\ \mathbf{h}(\mathbf{x}_K) \end{bmatrix} = \begin{bmatrix} h_1(\mathbf{x}_1) \cdots h_L(\mathbf{x}_1) \\ \vdots \quad \vdots \quad \vdots \\ h_1(\mathbf{x}_K) \cdots h_L(\mathbf{x}_K) \end{bmatrix}_{K \times L} \tag{4}$$

$$\mathbf{B} = \left[\boldsymbol{\beta}_1 \cdots \boldsymbol{\beta}_m\right]_{L \times m}, \text{ and } \mathbf{T} = \begin{bmatrix} \mathbf{t}_1 \\ \vdots \\ \mathbf{t}_K \end{bmatrix}_{K \times m}. \tag{5}$$

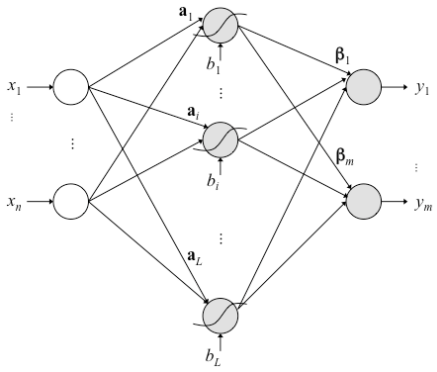

Figure 1. Topology of the neural network used in ELM.

Then, the matrix of output weights is

$$\mathbf{B} = \mathbf{H}^{-1}\mathbf{T}, \tag{6}$$

where $\mathbf{H}^{-1}$ is the generalized Moore-Penrose inverse of matrix $\mathbf{H}$.

B. Driving Database, Signals, and Features

The data collection used in this work was supplied by the "Drive-Safe Consortium" in Turkey. It was collected in Istanbul with a instrumented car equipped with different sensors [11]. We have used a subset of the database with $S = 11$ different drivers.

The car route is around 25 km (about 40 minutes), and includes different kinds of sections: city, very busy city, highway, highway with less traffic, a university campus, etc. The car route is the same for all drivers. However, the lengths of the driving sessions differ depending on the road conditions (e.g., traffic, weather).

The complete data set includes audio and video recordings, CAN-bus signals, gas pedal and break pedal sensor recordings, a frontal laser scanner, and an inertial measurement unit (IMU) with XYZ accelerometers and measures of rotation rates (pitch, roll and yaw). We used a reduced set of 14 signals that results from various considerations [8]. First, neither video nor audio signals were considered because of their intrusive nature. Then, a simple correlation study detected groups of signals with strong linear relationships. Only one representative signal from each group was selected. Other signals were discarded due to poor precision or for being highly influenced by environment infrastructure and topography.

Most of the signals were sampled at 32 Hz. The data of the rest of the signals, recorded at lower frequencies, were resampled to 32 Hz.

The data corresponding to the reduced set of 14 signals was processed to obtain a collection of informative features, namely the temporal means, the energies in the frequency domain, and one of the main coefficients in the cepstral domain, giving rise to a set of 42 variables. These features were calculated over frames of 128 seconds (4096 samples) for every second (32 samples).

III. CLOSED-SET DRIVER IDENTIFICATION MODEL

A *closed-set* driver identification model based on the ELM architecture was introduced in [8]. In [12], the model was optimized by using a multi-objective genetic algorithm that procures, at the same time, a reduced number of input variables, and a low number of hidden neurons in the ELM network. This method performs an intensive searching throughout the parameter space providing a set of Pareto optimal solutions. These solutions show the relevant driving variables for identification. The most suitable solution is finally selected based on the experience and the design requirements, such as those deriving from a digital HW implementation. The optimization was carried out according to the ability of the model to recognize the driver identity within the whole collection of 11 drivers.

The model, selected according to the multi-objective optimization, is a network with 32 hidden neurons working with the following set of 8 input variables: temporal features of gas pedal pressure (GP), roll rate (RR) and pitch rate (PR); frequency domain features of gas pedal and break (BP) pedal; and cepstral features of gas pedal, break pedal and steering wheel relative speed (SWRS) from de CAN-bus. Based on these parameters, we present a closed-set driver identification model for groups of m genuine drivers, i.e., an ELM network with topology 8-32-m. The network is depicted in Fig. 2.

The model has been tested with the whole set of 11 drivers, for whom it was optimized, as well as within subgroups of m = 2, 3, 4 and 5 drivers - similar to a real-life scenario to be used as a component of a smart car. These categories, determined by the number of drivers in the groups, are hereinafter referred to as 11d, 2d, 3d, 4d, and 5d for simplicity. Performance was measured as the Driver Identification Rate (DIR). This percentage was balanced to take into account the differences in the number of sample data among driving sessions due to the road conditions.

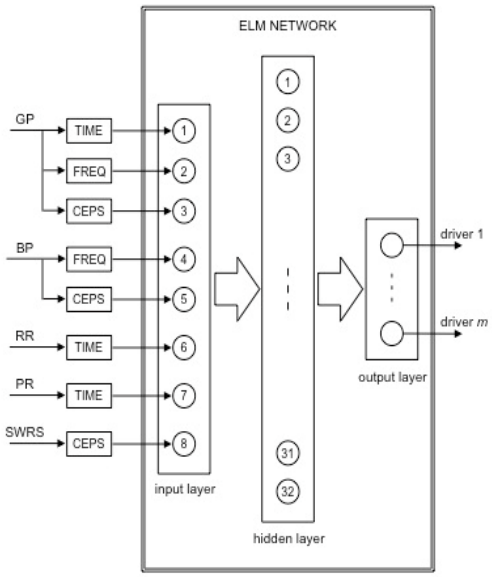

Figure 2. Architecture of the ELM network for the optimized closed-set driver identification system.

The test results in Table I correspond to the average Driver Identification Rates in all possible subgroups for the five categories. In this evaluation, two thirds of the data were intended to train the system, and the remaining one third was saved for testing. This partition was made separately for each driver dataset. In the experiments presented in Section IV, the same partition has been considered to provide a comparative framework. As can be seen, the designed model provides driver identification rates greater than 90% for every group, except for the whole set of 11 drivers, for which 84% is obtained.

IV. OPEN-SET MODEL FOR IMPOSTOR DETECTION

The objective of the open-set classification system is to detect a driver outside a group of known drivers, and simultaneously to maintain a high classification performance within the group of genuine drivers. Thus, we look for a driver identification system that includes rejection.

The problem of rejection is typically solved with thresholding in classifiers based on class models. These models are built from either probability densities or from some defined distance to prototypes [13]. Regarding neural networks as classifiers, they rarely provide any approximation to the a posteriori class probabilities. The implementation of a reject option is in most cases linked to a supervised learning scheme with a training algorithm designed to take into account rejection [14], [15], or to the use of training patterns with uncertainty [16].

For the ELM driver identification system presented in Section III, the analysis of the outputs in several examples shows that thresholding is not a suitable technique in this case. The use of some threshold is not favorable for the ELM networks proposed in this section either.

The approach proposed in this section considers an additional class, a *reject class*; therefore, it is an *open-set* model. The network in Fig. 2 is modified to include an additional output corresponding to the impostor, i.e., the reject class. The system is first discussed considering the multi-class character of the problem. Then, the results are analyzed from the point of view of group verification, i.e., considering that the objective of the system is just to verify whether a driver is a genuine driver of the group or it is an impostor instead. As a verification module the model would be a second stage classification connected to the closed-set driver identifier.

A. Open-Set Identifier: Multi-class View

In this case, we add a new output neuron to the classifying network in Fig. 2, i.e. an output for the reject class, the output $(m + 1)$ representing any driver other than those in a certain group. The topology of the new network is 8-32-(m+1). This system should find out whether a driver is an impostor or not. In addition, if the driver is a genuine one, the system should be capable of identifying the driver.

TABLE I. DRIVER IDENTIFICATION RATES OF CLOSED-SET MODEL

	2 drivers	**3 drivers**	**4 drivers**	**5 drivers**	**11 drivers**
No. Classifier	55	165	330	462	1
DIR	96.95	94.87	92.99	91.30	84.36

Each experiment considers a group of *m genuine drivers*. Another driver for each test case is marked as the *impostor* and excluded from training. The network is trained with data of the *m* genuine drivers of the group, and data from the rest of the drivers in the collection, which are labeled as *others*. In order to equally favor both the detection of the impostor and the genuine driver identification, we take as many training data of the drivers from the group as of the other drivers.

The model has been tested for groups with 2, 3, 4, and 5 drivers. For each group, formed from the collection of *S* drivers, there are $(S - m - 1)$ possible impostors. This multiplies the number of tests for each category to $\binom{S}{m} \cdot (S - m - 1)$.

For testing, the system computes the prediction on the basis of unseen data: testing data of the genuine drivers and data of the driver labeled as impostor, who is unknown for the classifier. Success is labeled as True Accept (TA) for a genuine driver correctly identified, and as True Reject (TR) for an outside driver correctly rejected. On the other hand, three types of error may occur, as shown in Table II for a group of three drivers: False Accept (FA), when the impostor is accepted as one of the genuine drivers; False Reject (FR), when a genuine driver is rejected; and misclassification (mc), if a genuine driver is rightly accepted as a driver of the group, but not correctly identified.

Viewed as a table of confusion, labels in Table II, i.e. *TA*, *TR*, *FA*, *FR*, and *mc*, represent the number of instances of every type of success and every type of error. For our multi-class classification problem, we add sub-indices to distinguish the results for the different genuine drivers of the groups. Misclassification instances can be grouped for each target driver, for example for a 3d group of drivers (1, 2, 3), $mc_2 = mc_{21} + mc_{23}$. In the same way, for the same example False Accept instances can be grouped as $FA = FA_1 + FA_2 + FA_3$.

Performance is firstly measured in terms of balanced hit-rates, separately for the impostor and for the drivers in the group. The Impostor Detection Rate (IDR) is the ratio of the number of testing data truly predicted as data of the impostor to the total number of testing data of the impostor. The Driver Identification Rate (DIR) is the mean of the identification rates for every driver in the group. These rates are expressed as follows:

$$\textit{Impostor Detection Rate, IDR} = \frac{TR}{TR + FA} \tag{7}$$

$$\textit{individual Driver Identification Rate, } iDIR_j = \frac{TA_j}{TA_j + FR_j + mc_j} \tag{8}$$

$$\textit{Driver Identification Rate, DIR} = \frac{1}{m}\sum_{j=1}^{m} iDIR_j \tag{9}$$

TABLE II. TYPE OF TEST RESULTS OF THE OPEN-SET DRIVER IDENTFICATION MODEL FOR GROUPS OF THREE DRIVERS

		Actual			
		driver 1	***driver 2***	***driver 3***	***impostor***
Prediction	***driver 1***	TA_1	mc_{21}	mc_{31}	FA_1
	driver 2	mc_{12}	TA_2	mc_{32}	FA_2
	driver 3	mc_{13}	mc_{23}	TA_3	FA_3
	impostor	FR_1	FR_2	FR_3	TR

Fig. 3 shows the impostor detection rate and the genuine driver identification rate, computed as the average rates for all tests (subgroup-impostor) in each category. In this bar chart, a loss of driver identification rate is visible with respect to the results of the closed-set identification system presented in Section III, which are outlined in red. However, the driver identification rates are greater than 80% for all the categories, and still above 90% for groups of 2 and 3 drivers. The impostor detection rate is above 80% when the car has a single genuine driver. This rate decays in inverse proportion to the number of authorized drivers, but it is greater than 50% in all cases. Table III presents the values of the ratios depicted in Fig. 3.

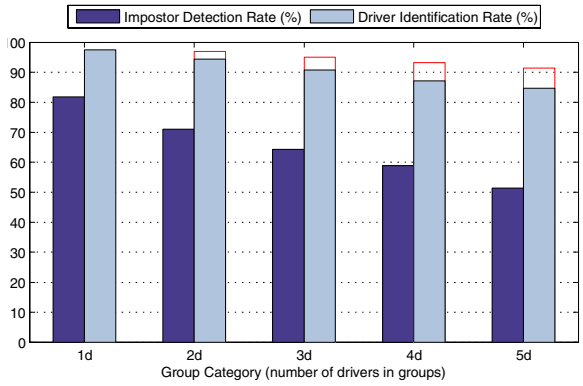

Figure 3. Average impostor detection rate and driver identification rate obtained with the open-set model for different group categories. Red line indicates the identufcation rates procured with the closed-set model.

TABLE III. PERFORMANCE RATES OF THE OPEN SET DRIVER IDENTIFICATION MODEL FOR DIFFERENT GROUP CATEGORIES

	1 driver	**2 drivers**	**3 drivers**	**4 drivers**	**5 drivers**
No. Classifiers	110	495	1320	2310	2772
IDR	81.22	70.01	63.13	58.67	52.99
DIR	97.75	94.40	90.61	86.80	84.79
IDR / DIR Accuracy	89.49	82.20	76.87	72.73	68.89
GIR	97.75	95.57	93.14	90.01	89.04
IDR / GIR Accuracy	89.49	82.79	78.13	74.34	71.01

B. *Open-Set Identifier: Group Verification View*

The same system, with network topology 8-32-(m+1), is evaluated from a binary perspective to assess its ability to distinguish whether a driver is a genuine driver of a given group or not. Hence, we add a new performance measure, the Group Identification Rate (GIR), to formulate the ratio of the number of testing data truly predicted as data of any genuine driver to the total number of testing data of genuine drivers of the group:

$$\textit{Group Identification Rate, GIR} = \frac{1}{m}\sum_{j=1}^{m}\frac{TA_j + mc_j}{TA_j + FR_j + mc_j} \quad (10)$$

In order to have a joint view of the individual behavior of the classifiers in each category, as well as the performance dispersion, we use a Receiver Operating Characteristic (ROC) diagram. We choose negative measures to represent each case on the ROC chart: the False Accept Rate (FAR) is the proportion of testing data for which the impostor is wrongly accepted as a genuine driver; the False Reject Rate (FRR) refers to the number of data for which a genuine driver is rejected as a driver of the group. These rates are expressed as follows:

$$\textit{False Accept Rate, FAR (FAR = 1 – IDR)} = \frac{FA}{TR + FA} \quad (11)$$

$$\textit{individual False Reject Rate, iFRR}_j = \frac{FR_j}{TA_j + FR_j + mc_j} \quad (12)$$

$$\textit{False Reject Rate, FRR (FRR = 1 – GIR)} = \frac{1}{m}\sum_{j=1}^{m} iFRR_j \quad (13)$$

The pair (FRR, FAR) resulting from every test is represented as a point on the ROC diagram. Fig. 4 shows the ROC diagram for all the tests corresponding to the category of 3 drivers.

Each classifier has been assigned a numerical quantity, a *quality score* that was defined according to its position on the ROC diagram. Specifically, we have considered the position of each point relative to the diagonals of the ROC space, depicted in Fig. 5. The diagonal D_1, from the left bottom corner to right top corner, is the line of *equal error rate*, FAR = FRR, i.e., the line of random classifiers. Points above this diagonal are better classifiers than points below the line. Therefore, we compute the distance of each point to D_1, and based on this measure, we assign a value in the range [0,1] to each point:

$$d_1^N = \frac{|x - y - 1|}{2}. \quad (14)$$

Point A corresponds to the best performance and it is assigned a $d_1^N = 1$. In the opposite corner of the graph, point D represents the case with worst performance. Any two points that are symmetrical with respect to the other diagonal, D_2, have the same value of d_1^N. However, considering

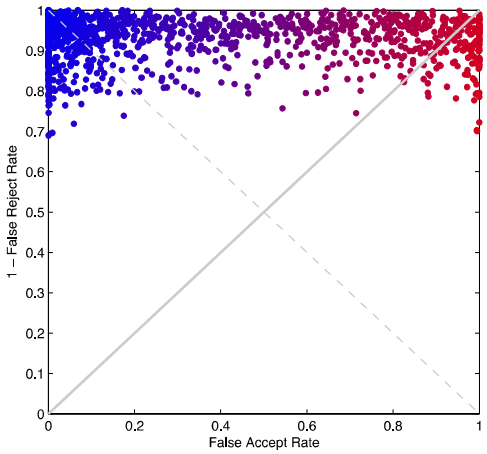

Figure 4. Performance representation of the test cases corresponding to the group category of 3 drivers on a ROC diagram.

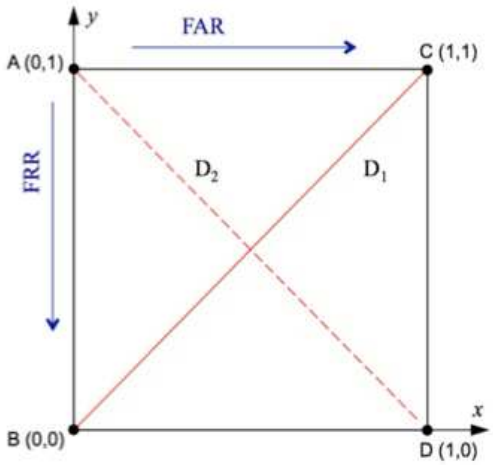

Figure 5. ROC space FAR/FRR.

the context of the problem, the two points have different significances. Let us see the extreme cases of points B and C on this line. The identifier represented by B never accepts the impostor as a genuine driver, but it always rejects the genuine drivers. Therefore, the system could be constantly warning the driver. In the case C, the genuine drivers are never rejected, but the impostor is never detected either. While B would lead to a somewhat annoying situation, C would completely fail as a security system.

In the definition of the quality score, we have considered the position relative to diagonal D_2 to take into account the contextual information described above. The points above diagonal D_2 will be marked as worst classifiers than those below the line. Therefore, we compute the distance of each point to D_2 and, based on this measure, we assign a penalty to those cases that are above the line:

$$d_2^c = l \cdot |x + y - 1|$$
$$l = \begin{cases} <1 \ \textit{if} \ FAR > FRR \\ 0 \ \textit{if} \ FAR < FRR \end{cases}. \quad (15)$$

Finally, the quality score, Q, is defined as:

$$Q = d_1^N - d_2^c. \tag{16}$$

The quality scores of points A, D and B are 1, 0, and 0.5, respectively. The parameter l in (15) representing the loss of quality is set to 0.35, so that the quality score of point C is lowered to 0.15. In Fig. 4, color is used according to the quality score of the model in each test case, from pure blue for the point of the best case to pure red for the worst case.

The distribution of quality Q among test cases in category 3d are shown in Fig. 6. It can be seen here that 44% of all the 3d classifiers tested show a good quality with $Q > 0.85$. However, as it can be observed for groups of 3 drivers in Fig. 6, and also in Fig. 4, there is a significant dispersion of performance among all the test cases within each group category.

In Table III, the performance rates corresponding to the group verification view are shown together with the rates of the multi-class view.

Another ELM model was investigated to work as a group verification system. In this case, a network with topology 8-32-2 was used to recognize two driving classes, the *group class* and the *reject class* or impostor. The test cases and the training data sets were identical to those in the model described previously in this section. Likewise, the analysis of the results was based on the table of confusion and on the quality score defined above. This approach resulted in slightly lower performances than the model with topology 8-32-(m+1). Fig. 7 shows the impostor detection rate and the group identification rate, with the ratios of the system with topology 8-32-(m+1) outlined in red for comparison.

V. Conclusions

A driver identification model with impostor detection is proposed. This module is an essential part for individualized driving assistant systems. The system is built on an extreme learning machine network, to take advantage of its very fast and simple training procedure. The model was designed based on a closed-set version, optimized to procure a high performance and a simple ELM network topology.

The system keeps good driver identification rates while it is capable of detecting impostors to some degree. The same scheme has been tested as a group verification model. In this case, the system would operate as a second stage following a closed-set classifier to provide driver identification.

The results show that modeling driver identification modules with impostor detection using only driving behavior signals does not procure a component reliable enough for smart cars. In future work, approaches that combine the information coming from systems based on driving behavior signals and systems using audio information will be investigated. In these future schemes, impostor detection for different decision window sizes will be also analyzed.

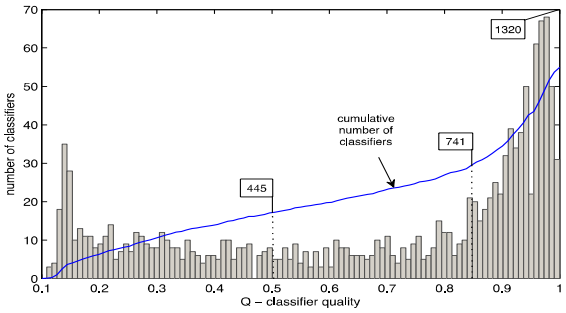

Figure 6. Quality distribution of test cases within the category of groups of 3 drivers. Blue line represents, for each Q_0 value, the cumulative number of classifiers with a quality $Q \le Q_0$.

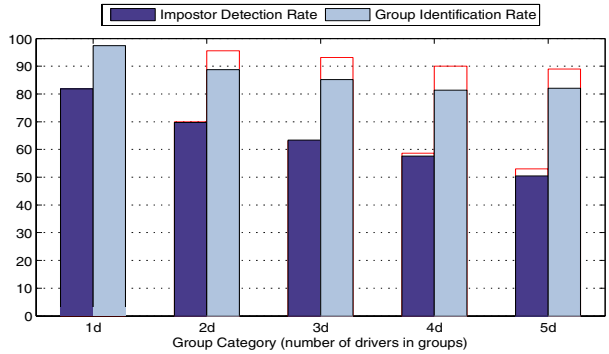

Figure 7. Performance of the ELM group verification system with topology 8-32-2.

Acknowledgment

The authors would like to thank Dr Hüseyin Abut and the researchers of the Drive-Safe Consortium in Istanbul (Turkey) for providing the "Uyanik" data set used to perform the experimentation.

References

[1] Castel, China Aerospace Telecommunications, "Smart Driver Behaviour Reader", http://www.castelecom.com/obd-gps-tracker (accessed June 2016).

[2] J. Stallkamp, H. K. Ekenel, H. Erdogan, R. Stiefelhagen, and A. Erçil, "Video-based driver identification using local appearance face recognition", *Workshop DSP in Mobile and Vehicular Syst.*, Istanbul, Turkey, June 2007.

[3] J.-D. Wu, and S.-H. Ye, "Driver identification based on voice signal using continuous wavelet transform and artificial network techniques", *Expert Systems with Applications*, vol. 36, p. 1061-1069, 2009.

[4] C. Craye, "A framework for context-aware driver status assessment systems", thesis of Master of Appl. Science in Electr. Comp. Engin., Waterloo Univ., Waterloo, Ontario, Canada, 2013.

[5] A. Riener, and A. Fersha, "Supporting implicit human-to-vehicle interaction: driver identification from sitting postures", The First Annual Int. Symp. On Vehicular Computing Systems, Dublin, Ireland, 22-24 July, 2008.

[6] E. Öztürk, and E. Erzin, "Driver Status Identification from Driving Behavior Signals", in *Digital Signal Processing for In-vehicle Systems and Safety*, J.H.L. Hansen, P. Boyraz, K. Takeda, and H. Abut, eds, Springer, 2012, ch.3.

[7] H. Qian, Y. Ou, X. Wu, X. Meng, and Y. Xu, "Support Vector Machine for Behavior-Based Driver Identification System," Journal of Robotics, vol. 2010, 11 pages, 2010.

[8] M. V. Martínez, I. del Campo, J. Echanobe, and K. Basterretxea, "Driving behavior signals and machine learning: a personalized Driver

assistance system", *IEEE Int. Conf. Intelligent Transportation Systems*, Las Palmas de Gran Canaria (Spain), September 15-18, 2015.
[9] C. K. Chow, "On optimum recognition error and reject tradeoff", *IEEE Trans. Information Theory*, vol. IT-16, pp. 41-46, 1970.
[10] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme Learning Machine for Regression and Multiclass Classification," IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics, vol. 42, no. 2, pp. 513-529, 2012.
[11] H. Abut, H. Erdogan, A. Ercil, et al., "Data collection with "UYANIK: too much pain; but gains are coming," in Corpus and Signal Processing for Driver Behavior, K. Takeda, J. H.L. Hansen, H. Erdogan, and H. Abut, (eds.) Springer Business-Science, 2008, ch3.
[12] J. Echanobe, I. del Campo, and M. V. Martínez, "Design and optimization of a neural network-based driver recognition system by means of a multiobjective genetic algorithm", IEEE World Congress on Computational Intelligence, Vancouver (Canada), July 24-29, 2016.
[13] T. C. W. Landgrebe, D. M. J. Tax, P. Paclik, and R. P. W. Duin, "The interaction between classification and reject performance for distance-based reject-option classifiers", *Pattern Recognition Letters*, vol. 27, pp. 908-917, 2005.
[14] D. M. J. Tax, and R. P. W. Duin, "Growing a multi-class classifier with reject option", *Pattern Recognition Letters*, vol. 29, pp. 1565-1570, 2008.
[15] R. Sousa, A. R. da Rocha Neto, J. S. Cardoso, and G. A. Barreto, "Classification with reject option using the self-organizing map", Artificial Neural Networks and Machine Learning – ICANN 2014, pp. 105-112, 2014.
[16] H.Ishibuchi, and M. Nii, "Neural networks for soft decision-making", *Fuzzy Sets and Systems*, vol. 115, pp. 121-140, 2000.