Using Hidden Markov Models for Profiling Driver Behavior Patterns

Daniel Nieto, Andres Giraldo, Eduardo Giraldo, Juan Martínez, Luis Trujillo, Yohan Céspedes, Juan Acosta

***Abstract*—Detection of risky driving events using smartphone based sensing is a growing technology devoted to impact positively driving behaviors. This technology might improve traffic and reduce the number of car accidents. However, data measured from inbuilt smartphone sensors, represented as a multivariate time series, commonly contains strong temporal dynamics. As a result, there is a growing need for developing methods able to handle such dynamics to make any inference based on the data in hand. In this work, we present a methodology for discriminating risky driving events from smartphone sensors using Hidden Markov Models, which are a well known statistical method for dealing with time-series comprising time-varying information. The methodology is validated using a publicly available dataset, where we demonstrated that the achieved results are comparable with state-of-the-art approaches, yielding accuracy rates around 90% in a seven classes problem.**

***Index Terms*—Hidden Markov Models, Driver behavior, Smartphone sensors.**

I. Introduction

Nowadays, given the uncontrolled population and traffic volume growth, several cities around the world have been forced to install Intelligent Transportation Systems (ITSs) ntended to reduce problems that deteriorate life quality in large cities as traffic jams [1], feeder bus routes [2]. [3], CO_2 emissions, emergencies, and accidents [4]. One of the variables that might help to mitigate such problems is the driver behavior pattern. In turn, several studies have demonstrated that under continuous monitoring, drivers tend to reduce dangerous and aggressive conducts, reducing as well the number of accidents on the roads [5]. As a result, there is a growing need for developing driver behavior classification systems that feedback motorists by rewarding or penalizing their driving patterns within a specific time interval.

One of the biggest challenges for designing ITSs is the sensing technology needed to collect data from moving vehicles, divided into intra vehicular and urban sensing platforms [6]. Within the first class, some transportation companies equip their vehicles with GPS sensors and cameras [7]. However, the high cost of such sensors limits their use. In the second class, several cities have installed surveillance cameras and velocity sensors in their most traveled roads [8]. Nevertheless, the coverage of such systems is quite limited in large cities. Moreover, smartphone based sensing systems appear as a low cost solution that can be used both for companies, cities, and even regular persons willing to improve their driving performance. Among their many advantages, smartphone-based systems are scalable, upgradeable, and the analysis can be performed in real time [9]. Consequently, smartphone based telematics systems for profiling driver behavior, mostly based on accelerometers, gyroscopes, and GPS, are gaining increasing attention [10].

Several approaches based on smartphones have been proposed to analyze the information provided by their sensors, as vehicle locations [11], car following models [1], and driver behavior, obtaining data even by pedestrians and cyclists [12]. Most of them attempt to identify risky driving maneuvers to categorize motorists into predefined risk levels. The most used methodology consists in applying sliding windows to segments of sensor data, to later calculate several features from each window. Finally, those features are used to feed either a fuzzy or a machine learning algorithm to discriminate among the studied driving events [13], [14], [15], [16]. However, the selection of features based on sliding windows requires human expertise to preserve the strong temporal nature of smartphone sensor data. Furthermore, fuzzy logic tends to be subjective and sensitive to noise as it requires set several thresholds. Regarding machine learning algorithms, their performance varies from work to work as they highly depend on the set of estimated features and their parameters, like window size, and sample rate. To avoid feature extraction procedures, several works intend using raw sensor data to feed deep neural networks to identify risky driving events [17], [18]. However, the success of deep neural networks largely depends on high information volumes and a proper architecture selection.

In this study, we propose using Hidden Markov models (HMMs) to classify driving maneuvers only based on a smartphone accelerometer and gyroscope. As an advantage, HMMs exploit the time varying nature of sensor data, making it an ideal tool for this specific task. Specifically, we first train an HMM model for each desired event. Later, we evaluate the testing segments on each pre trained model and assign a label according to the highest reached probability. We also explore which combination of analyzed sensors produce the best driving event classification. Furthermore,

Manuscript received December 03, 2019; revised October 03, 2020.

D. Nieto is with the Instituto Tecnológico Metropolitano, Medellín, Colombia. Email: danielnieto152326@correo.itm.edu.co.

A. Giraldo is with the Instituto Tecnológico Metropolitano, Medellín, Colombia. Email: andresgiraldo@itm.edu.co.

E. Giraldo is with the Universidad Tecnológica de Pereira, Pereira, Colombia. Email: egiraldos@utp.edu.co

J. Martínez is with the Instituto Tecnológico Metropolitano, Medellín, Colombia. Email: juanmartinez@itm.edu.co

L. Trujillo is with the Centro de Excelencia y Apropiación en Internet de las Cosas (CEA-IoT), Bogotá, Colombia and Pontificia Universidad Javeriana, Bogotá, Colombia. Email: trujillo.luis@javeriana.edu.co

Y. Céspedes is with the Centro de Bioinformática y Biología Computacional de Colombia - BIOS, Manizales, Colombia and Universidad Autónoma, Bogotá, Colombia. Email: yohan.cespedes@bios.co

J. Acosta is with the Centro de Bioinformática y Biología Computacional de Colombia - BIOS, Bogotá, Colombia. Email: acosta.juanc@javeriana.edu.co.

we compare our HMM based approach against i) short time features and machine learning based classification, and ii) raw data and deep learning based classification.

II. Methods

A. Hidden Markov Models

HMMs assume that measured data, in this case, smartphone sensor data, are generated from hidden discrete states. Moreover, associated with each state exists an observation model mapping probabilistically each state to the observed data [19].

Thus, let $\mathcal{X} = \{\mathbf{X}_i \in \Re^{C \times T_i},\ i = 1, \ldots, N\}$ a set of N observations belonging to an specific driving event, e.g, sudden acceleration, and consider that each observation $\mathbf{X}_i$ was measured in C sensors at T_i time samples. We assume for each observation a HMM of length T_i, state space dimension K, and hidden state variables $\mathbf{S} = \{\mathbf{s}_1, \ldots, \mathbf{s}_{T_i}\}$. The full posterior probability of the model is given by:

$$P(\mathbf{X}_i, \mathbf{S}) = P(\mathbf{s}_0|\pi_0) \prod_{t=1}^{T_i} P(\mathbf{s}_t|\mathbf{s}_{t-1}, \mathbf{\Pi}) P(\mathbf{x}_{it}|\mathbf{s}_t) \qquad (1)$$

where $\pi_0 \in \Re^K$ is the initial state probability, $\mathbf{\Pi} \in \Re^{K \times K}$ is the transition probability matrix where its (k, j) element describes the transition probability from state k to state j between time $t-1$ and t. $P(\mathbf{x}_t|\mathbf{s}_t, \mathbf{\Theta})$ is the observation model describing the data distribution for each state, with $\mathbf{x}_{it} \in \Re^C$ the measurements of all sensors at time instant t. Here, we assume that the observation model for each state is a multivariate Normal Distribution (MVN) with parameters $\mathbf{\Theta}_k = \{\mu_k, \mathbf{\Sigma}_k\}$ where $\mu_k \in \Re^K$ is the mean vector, and $\mathbf{\Sigma}_k \in \Re^{K \times K}$ is the covariance matrix.

The inference of the model parameters for each observaion sequence $\lambda_i = \{\pi_0, \mathbf{\Pi}, \mathbf{\Theta}_k\}$ is carried out using an iterative Expectation Maximization (EM) algorithm, known as the Baum Welch algorithm.

Moreover, in presence of N different observation sequences for each driving event, the full probabilistic model is given by

$$P(\mathcal{X}|\lambda) = \prod_{i=1}^{N} P(\mathbf{X}_i|\lambda) = \prod_{i=1}^{N} P_i. \qquad (2)$$

This modification simplifies the model for estimating a single set of parameters λ for each class [20].

III. Experiments

A. Analized data

The data used in this work was collected during four car trips of approximately 13 minutes. The smartphone used was a Motorola XT1058 with Android version 5.1. For reducing the measurement noise, during the entire trip, the smartphone was fixed to the car windshield. Also, weather conditions were sunny and the asphalt was dry. The data is freely available at [1]. During the trips, seven different driving events were labeled, which are summarized along with their occurrence in table I.

Driving event	Occurence
Aggressive breaking	12
Aggressive acceleration	12
Aggressive left turn	11
Aggressive right turn	11
Aggressive left lane change	4
Aggressive right lane change	5
Non-aggressive event	14
Total	**69**

TABLE I
Summary of considered driving events along with their occurrence.

Before the data analysis, sensor measurements were rotated from the device coordinates system to the standard earth coordinates. As a result, the analyzed data include two tridimensional ($x, y,$ and z) time series taken from the accelerometer and the gyroscope, each recorded with a sample rate of 50 Hz.

1) Training and Validation of the HMM based identification: Identification of risky driving events based on HMMs consists of two main stages, namely, training and validation. In the training stage, we used all the segments belonging to a particular driving event for learning the models of an HMM, as shown in Fig. 1.

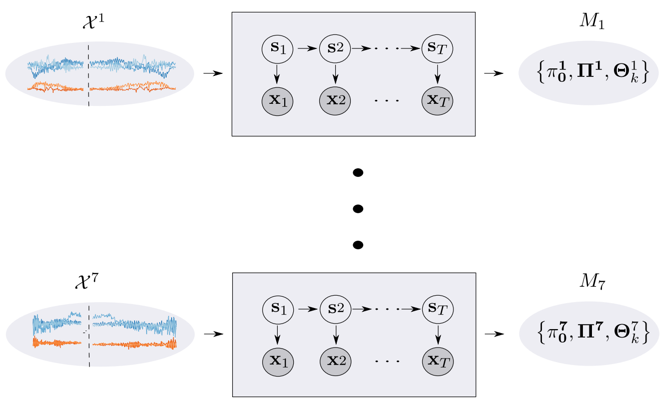

Fig. 1. HMM trainin scenario: A model is trained with all the segments belonging to an specific class.

As an advantage, it is worth noticing that the HMM training does not require that all the segments have the same length. To simplify the training stage, we assume diagonal covariance matrices to reduce the number of parameters to estimate. As a result, in this stage, we obtain seven HMM models (one per class).

Later, in the validation stage, when a non labeled segment arrives, using the forward/backward algorithm, we evaluated the probability that such segment was generated for each of the trained models, and we assigned to such segment the label of the model that produced the highest probability, as shown in Fig. 2. As in the training procedure, the evaluation was carried out regardless of the segment length.

[1]https://github.com/jair-jr/driverBehaviorDataset

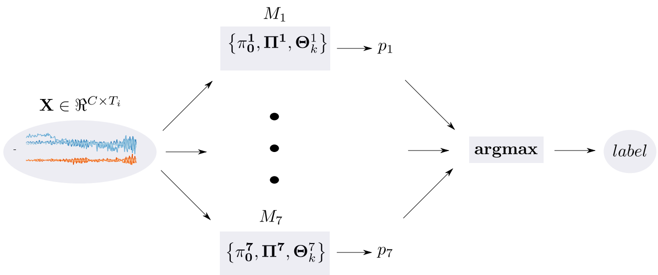

Fig. 2. HMM testing scenario: An unlabeled segment is evaluated in each model. The assigned label corresponds to the model that generates the highest probability of generating the testing segment.

For selecting the training and testing sets, we used 5 folds stratified cross validation with 70% of the data for training and the remaining 30% for validation. We selected this strategy to ensure that in each fold, the training and testing sets comprised segments belonging to all the considered classes in almost the same proportion.

Here, we analyzed three parameters that influence the performance of the proposed classification framework, namely, i) the number of HMM states, ii) the interval of each segment, and iii) the combination of sensor axes.

IV. Results

A. Number of HMM states

In the first step, we adjusted the optimal number of HMM states used for the classification of risky driving events. The experiment was carried out independently for the accelerometer and the gyroscope, using the three dimensional time series corresponding to the x, y, and z axes. The number of states was set from 1 to 9. Achieved results are shown in Fig. 3.

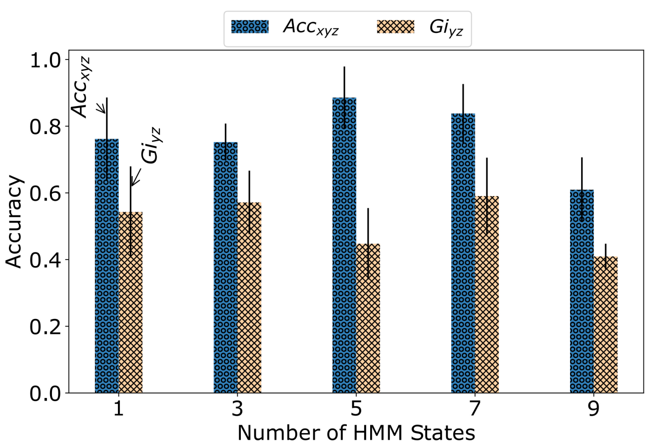

Fig. 3. Tuning the number of HMM states.

It is clear that for the accelerometer, the optimum number of parameters is 5. However, there is not a clear difference among the classification accuracies achieved by the gyroscope for all tested number of HMM states. Consequently, for the gyroscope, we set this parameter as 3.

B. Interval of analyzed segments

In the database, considered driving events are marked from an initial to a final time point t_i and t_f respectively, as shown in Fig. 4.

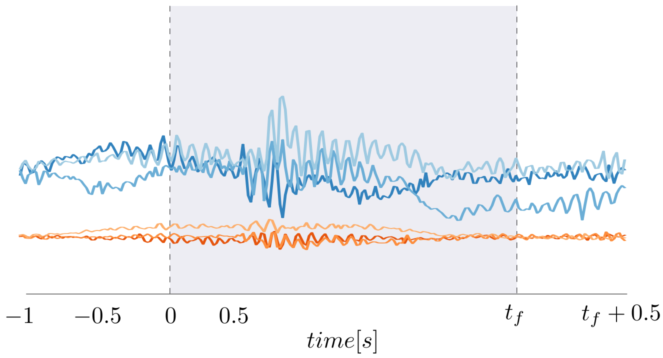

Fig. 4. Example of a segment with the initial t_i and the final t_f points. Furthermore, the remaining starting points at $t_i - 0.5$ and $t_i - 1$, and final points at $t_i + 0.5$, $t_f + 1$ are shown.

However, to include the transient between regular to risky driving in the analysis, we considered, besides the initial time t_i, starting points in 0.5 and 1 second before t_i. Also, we considered three different ending points to see if the segment length affects the performance of the HMM. These points were 0.5 seconds after the event has started, the marked ending time, and 0.5 seconds after the marked ending time. Summarizing, we considered nine different lengths for each segment, resulting in nine possible combinations of the initial and final points. Details of the segments under consideration can be seen in table II

Segment ID	**Time**
seg_1	$[t_i - 1, t_i + 0.5]$
seg_2	$[t_i - 1, t_f]$
seg_3	$[t_i - 1, t_f + 0.5]$
seg_4	$[t_i - 0.5, t_i + 0.5]$
seg_5	$[t_i - 0.5, t_f]$
seg_6	$[t_i - 0.5, t_f + 0.5]$
seg_7	$[t_i, t_i + 0.5]$
seg_8	$[t_i, t_f]$
seg_9	$[t_i, t_f + 0.5]$

TABLE II
Summary of considered event segments.

In Fig. 5, we noticed that the longest the analyzed segment, the better the HMM performance. In turn, the worst result was achieved with the shortest segment. We also noticed that including a segment before t_i does not improve either worsen the solution. This can be explained seeing the Fig. 4, where it is noticeable that the event does not begin at the marked starting point. Thus, analyzing the entire segment already includes the transient period that would produce a hidden state transition in the HMM.

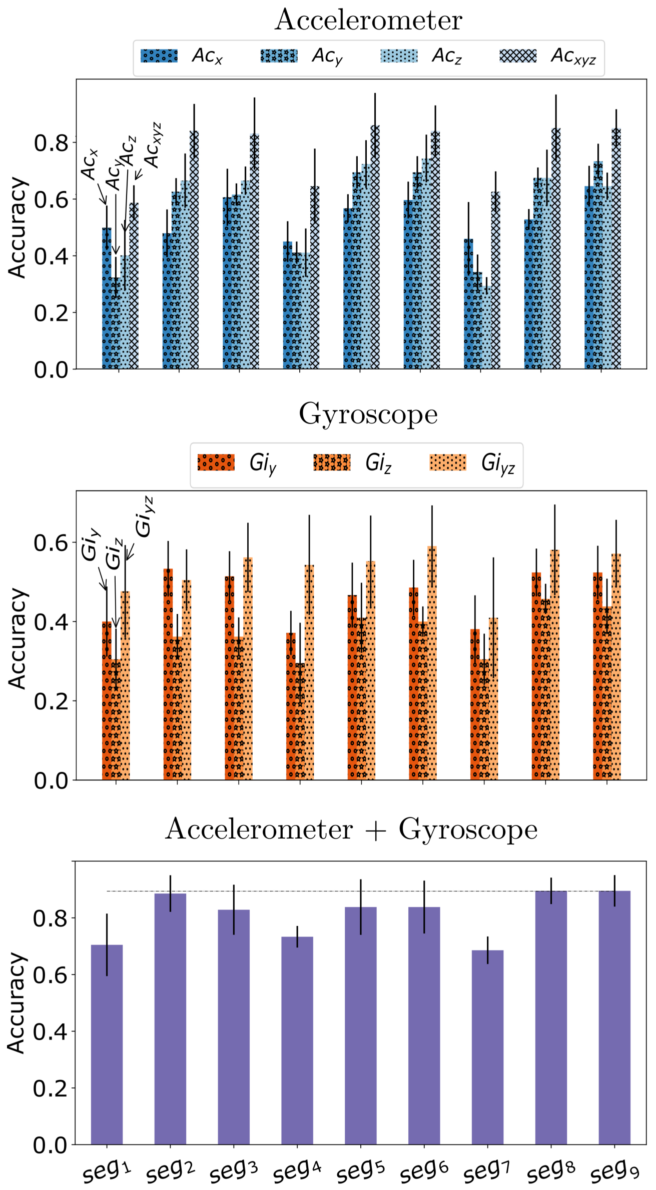

Fig. 5. Accuracy achieved by the HMM in the testing group, under all the possible combinations of sensores and segment longitudes.

C. Analyzed sensor

We trained the HMMs using separately each *x, y,* and *z* axis of each sensor (accelerometer and gyroscope). We also trained the HMMs using the three axes of the accelerometer, and the three axes of the gyroscope. Lastly, we trained using all the information from both sensors. We performed all the combinations of time segments and sensor axes for training and testing the HMM performance, in order to obtain its best configuration. Results are shown in Fig. 5. As expected, a single sensor was not able to properly identify all the considered events. This can be seen in the first three bars of the top and the middle figures. However, when mixing all the axis belonging to the same sensor (fourth bar of the same figures), the accuracy was dramatically improved. Moreover, when combining all the used sensors, the accuracy improved even more. As a result, we obtained that the more information used to feed the HMM model of each class, the more accurate the risky driving event classification. It is worth to notice that when all time series were mixed, the number of HMM states was set to 5.

1) Comparisson approaches: We compared the proposed HMM based classification against several state of the art methodologies for the detection of risky driving events. On the one hand, we compared with two different recursive neural networks (RNNs), namely, the Long Short Term Memory Network (LSTM) and Gated Recurrent Units (GNU), with the configuration described in [17]. Both RNNs were trained with 10 units, a softmax activation function, and RMSprop as the optimization algorithm. These networks were implemented using the Keras framework [2]. As the RNNs require segments with the same lengths, we cropped both training and testing observations to have the shortest length available in the database.

On the other hand, we compared with a sliding window characterization followed by several machine learning algorithms to classify the different driving events. As suggested in [14], we used four windows of one second, but including 0.25 seconds overlap. For each window, we computed several statistics, namely, mean, median, standard deviation, and mode. As a result, we obtained 16 features for each segment. Later, three different classifiers were trained: Support Vector Machines (SVMs) with a RBF kernel, Random Forest, and Neural Network.

The comparison with the state of the art methods was carried out using all the time series provided by the sensors within the labeled segments in the database. In all the comparison scenarios, we used the same cross validation scheme.

Figure 6 shows the obtained results of the comparisson described above. Besides the accuracy, we used as performance metric the area under the curve (auc) of each class, as suggested in [14], along with an averaged auc for all classes.

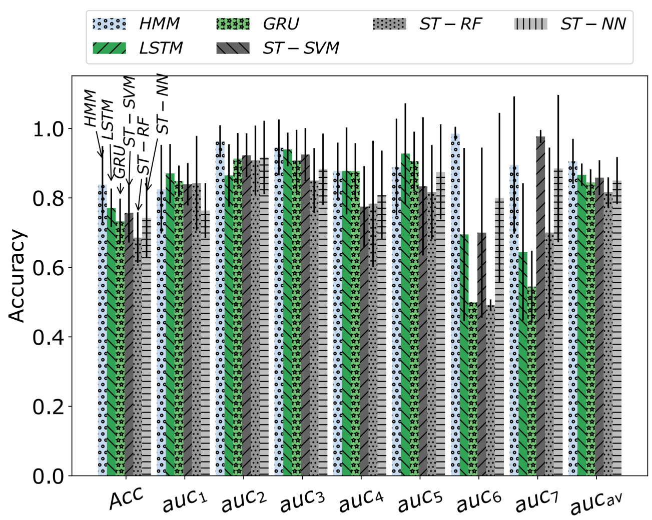

Fig. 6. Driving events comparison with methodologies from the state of the art.

[2]https://keras.io

Regarding the accuracy, the method based on HMM achieves the highest results with the lowest deviation among folds. However, this method presented difficulties discriminating events of class 1 (Aggressive breaking) and class 6 (Aggressive right lane change). The former situation could be explained seeing the class 2 AUC, close to 1. This might imply that sudden velocity changes can be confused. A similar phenomenon could be producing the Aggressive right lane change miss classification, as the AUC of aggressive right turn is also close to 1. In the remaining classes, the HMM produced either better or competitive results compared with the best state of the art method, in this case, short time features classified with NNs. A particular behavior is presented by the RNNs, yielding the lowest results. This might be explained by the simple configuration used or by the low number of training segments.

D. Analysis of real driving scenarios

We analyzed real driving scenarios. In this case, an HMM was trained with the labeled segments of three out of the four travels recorded in the data acquisition process. Then, segments lasting one second of the remaining travel were labeled according to the trained HMM as explained in section III-A1. Results for each travel are shown in Fig. 7, where check marks indicate segments that were properly labeled, whereas x-marks indicate badly labeled segments. Moreover, white font corresponds to non aggressive event assigned by our method. In all travels, it can be seen that non aggressive events correspond to areas of the time series where the amplitude of the measured data (accelerometer and gyroscope) presented a soft behavior. Furthermore, our method identified properly abrupt changes, and, in most cases, the true label corresponds with the assigned label.

E. Third party data

As the final experiment, we analyzed third party data. To this end, we trained HMMs with the four travels recorded in the data acquisition process. Then, we collected data under similar conditions to the used ones to record the database mentioned above, namely, the same frequency rate, the same sensors, and the same fixed location of the phone inside the car. Later, as in the previous experiment, we passed through the trained HMM models one second segments of the newly recorded data, and we labeled such segments according to the highest probability. Results are shown in Fig. 8, where blue lines indicate non aggressive events. It can be seen that our model identified aggressive right and left lanes on right or left curves on the road. Consequently, we can conclude that the model properly identifies risky driving events.

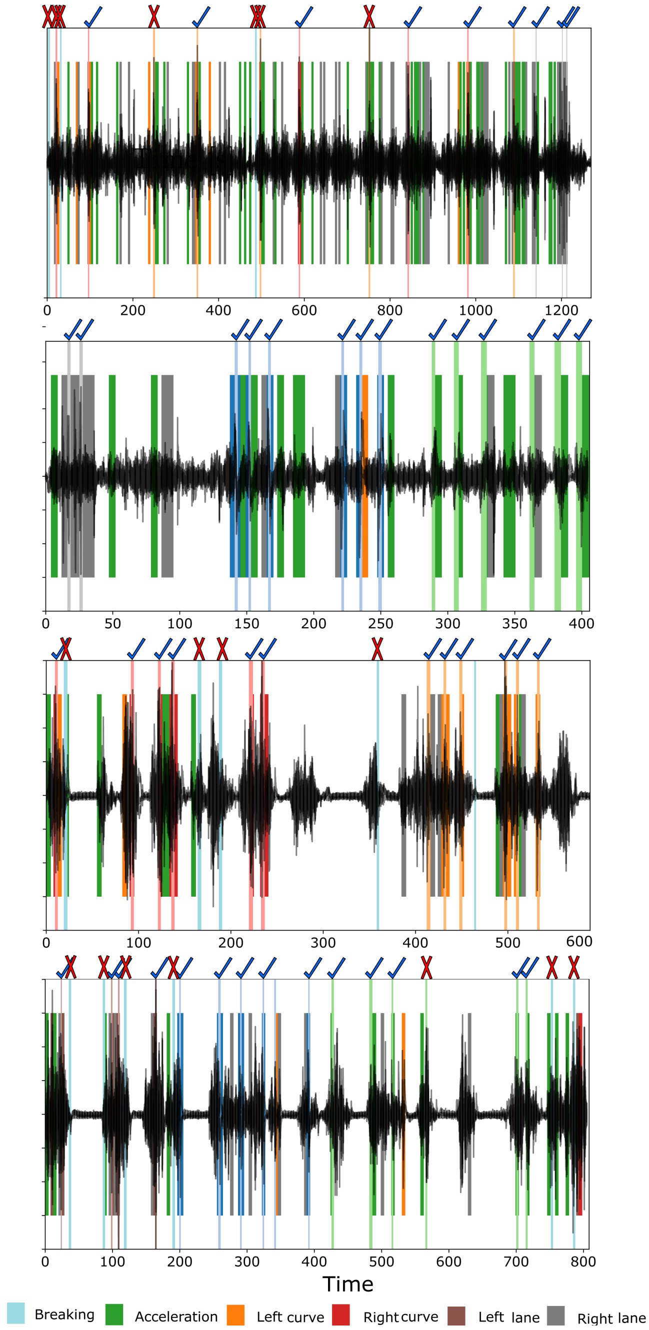

Fig. 7. Analysis of real driving scenarios.

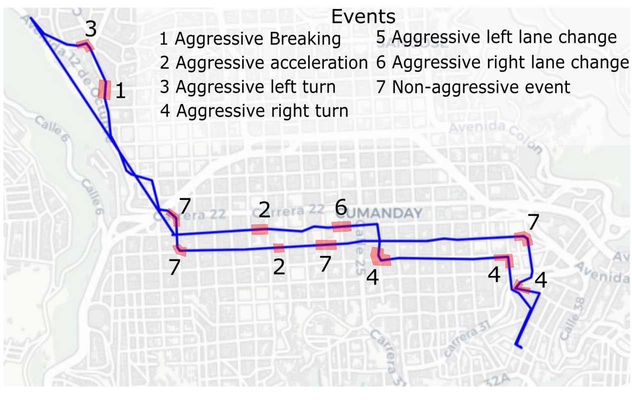

Fig. 8. Analysis of third party data.

V. Conclusions

In this work, we presented a methodology for detecting risky driving events based on HMMs using accelerometer and gyroscope data recorded from a smartphone. We studied which sensor configuration and what recorded segment produced the best accuracy results in detecting seven different types of driving events. We used HMM because of its intrinsic ability to deal with time series with strong temporal dynamics. We demonstrated that using as much information as possible, i.e., the six time series provided all the axes of both sensors, along with all the recorded observations, the classification accuracy is dramatically improved. As an extra benefit of our approach, there is no need for clipping the recordings to have the same length. Moreover, we obtained competitive or even better results compared to some state of the art methodologies.

As future work, we would like to test our approach with data recorded in different weather conditions and with different drivers, to validate if the designed system is worth to be used within an online driving monitoring platform. Moreover, we would like to expand the probabilistic model to consider within the estimation of the parameters a wider range of past values.

Acknowledgements

The authors would like to acknowledge the cooperation of all partners within the *Centro de Excelencia y Apropiación en Internet de las Cosas (CEA-IoT)* project. The authors would also like to thank all the institutions that supported this work: the Colombian Ministry for the Information and Communications Technology (*Ministerio de Tecnologías de la Información y las Comunicaciones - MinTIC*) and the Colombian Administrative Department of Science, Technology and Innovation (*Departamento Administrativo de Ciencia, Tecnología e Innovación - Colciencias*) through the *Fondo Nacional de Financiamiento para la Ciencia, la Tecnología y la Innovación Francisco José de Caldas* (Project ID: FP44842-502-2015). The authors would also like to thank the anonymous referees for their valuable comments and helpful suggestions.

References

[1] D. Wenju, L. Yinzhen, Z. Jiangang, and Y. Jianning, "Stability Analysis and Control of a Full Velocity Difference Car-following Model," *Engineering Letters*, vol. 28, no. 71861023, 2020.

[2] W. C. Ho, J. M. Su, and C. Y. Chang, "Maximal market potential of feeder bus route design using particle swarm optimization," *IAENG International Journal of Applied Mathematics*, vol. 49, no. 4, pp. 470–477, 2019.

[3] X. Tian, F. Ruan, H. Cheng, and Q. Zheng, "A signal timing model for improving traffic condition based on active priority control strategy," *Engineering Letters*, vol. 28, no. 1, pp. 235–242, 2020.

[4] J. Guerrero-Ibáñez, S. Zeadally, and J. Contreras-Castillo, "Sensor technologies for intelligent transportation systems," *Sensors*, vol. 18, no. 4, p. 1212, 2018.

[5] J. S. Hickman and E. S. Geller, "Self-management to increase safe driving among short-haul truck drivers," *Journal of Organizational Behavior Management*, vol. 23, no. 4, pp. 1–20, 2005.

[6] J. A. Guerrero-Ibanez, S. Zeadally, and J. Contreras-Castillo, "Integration challenges of intelligent transportation systems with connected vehicle, cloud computing, and internet of things technologies," *IEEE Wireless Communications*, vol. 22, no. 6, pp. 122–128, 2015.

[7] J. Bandeira, T. G. Almeida, A. J. Khattak, N. M. Rouphail, and M. C. Coelho, "Generating emissions information for route selection: Experimental monitoring and routes characterization," *Journal of Intelligent Transportation Systems*, vol. 17, no. 1, pp. 3–17, 2013.

[8] T. Idé, T. Katsuki, T. Morimura, and R. Morris, "City-wide traffic flow estimation from a limited number of low-quality cameras," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 4, pp. 950–959, 2016.

[9] G. Castignani, T. Derrmann, R. Frank, and T. Engel, "Driver behavior profiling using smartphones: A low-cost platform for driver monitoring," *IEEE Intelligent Transportation Systems Magazine*, vol. 7, no. 1, pp. 91–102, 2015.

[10] A. Chowdhury, T. Chakravarty, A. Ghose, T. Banerjee, and P. Balamuralidhar, "Investigations on driver unique identification from smartphoneâFIXME™s gps data alone," *Journal of Advanced Transportation*, vol. 2018, 2018.

[11] T. T. Zan and S. Phyu, "Mobile location based indexing of notification system," *Lecture Notes in Engineering and Computer Science: Proceedings of The International MultiConference of Engineers and Computer Scientists 2018, IMECS 2018,14-16 March, 2018, Hong Kong*, pp. 88–93.

[12] J. A. Castán, S. M. Ibarra, J. M. Laria, M. G. Treviño, J. C. Pérez, L. J. González, and E. R. Castán, "Intelligent ecosystems to increase urban traffic quality," *Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering 2017, WCE 2017, 5-7 July, 2017, London, U.K.*, pp. 221–225.

[13] J. Xie, A. R. Hilal, and D. Kulić, "Driving maneuver classification: A comparison of feature extraction methods," *IEEE Sensors Journal*, vol. 18, no. 12, pp. 4777–4784, 2017.

[14] J. F. Júnior, E. Carvalho, B. V. Ferreira, C. de Souza, Y. Suhara, A. Pentland, and G. Pessin, "Driver behavior profiling: An investigation with different smartphone sensors and machine learning," *PLoS one*, vol. 12, no. 4, p. e0174959, 2017.

[15] D.-N. Lu, D.-N. Nguyen, T.-H. Nguyen, and H.-N. Nguyen, "Vehicle mode and driving activity detection based on analyzing sensor data of smartphones," *Sensors*, vol. 18, no. 4, p. 1036, 2018.

[16] H. R. Eftekhari and M. Ghatee, "A similarity-based neuro-fuzzy modeling for driving behavior recognition applying fusion of smartphone sensors," *Journal of Intelligent Transportation Systems*, vol. 23, no. 1, pp. 72–83, 2019.

[17] E. Carvalho, B. V. Ferreira, J. Ferreira, C. de Souza, H. V. Carvalho, Y. Suhara, A. S. Pentland, and G. Pessin, "Exploiting the use of recurrent neural networks for driver behavior profiling," in *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2017, pp. 3016–3021.

[18] S. Hernández Sánchez, R. Fernández Pozo, and L. Hernández Gómez, "Estimating vehicle movement direction from smartphone accelerometers using deep neural networks," *Sensors*, vol. 18, no. 8, p. 2624, 2018.

[19] M. W. Woolrich, A. Baker, H. Luckhoo, H. Mohseni, G. Barnes, M. Brookes, and I. Rezek, "Dynamic state allocation for meg source reconstruction," *Neuroimage*, vol. 77, pp. 77–92, 2013.

[20] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.