Alcohol Detection over Long Periods Using Smartphone Accelerometer Data

Manuel Gil-Martín, Rubén San-Segundo, Cristina Luna-Jiménez
Speech Technology and Machine Learning Group, Information Processing and Telecommunications Center,
E.T.S.I. Telecomunicación, Universidad Politécnica de Madrid
Madrid, Spain
Email: manuel.gilmartin@upm.es; ruben.sansegundo@upm.es; cristina.lunaj@upm.es

***Abstract*—This paper proposes a motion biomarker for alcohol detection using a deep learning approach that processes inertial signals recorded with a smartphone. The deep learning architecture is composed of a Convolutional Neural Network, including three convolutional layers for learning features from the inertial signal spectrum, and several fully connected layers to perform classification and regression tasks. The motion biomarker is computed in two steps. Firstly, the inertial signals are segmented in short sub-windows (3-6 seconds) and the system generates a score for each sub-window. Secondly, the scores in consecutive sub-windows are combined to provide a motion biomarker over in longer periods of time (30 seconds). This paper compares the proposed approach to previous works using the same experimental dataset and setup: Bar Crawl Detecting Heavy Drinking Data Set, K-fold cross-validation methodology and two tasks (classification and regression). The proposed deep learning approach overperformed previous reported results: the accuracy increased 4 % (absolute) when classifying between intoxicated and sober participants and the Mean Squared Error relatively decreased 9 % when estimating the Transdermal Alcohol Content of the participants by averaging the scores from consecutive sub-windows.**

Keywords-Alcohol Detection; Motion Wearable Sensors; Convolutional Neural Networks; Sub-windows Combination.

I. INTRODUCTION

High-frequency alcohol consumption could become a serious threat for people's health. In fact, physicians and social workers are interested in reducing the alcohol consumption in young adults. For example, measuring the Transdermal Alcohol Content (TAC) is useful to recommend the person to stop drinking in real time. In addition, wearable technology could be used for developing an alcohol detection system based on inertial signals from accelerometers included in smartphones.

This paper evaluates a strategy to combine sub-windows information for an alcohol detection system based on wearable technology and deep learning, obtaining important improvements for long windows (over 10 seconds). This study was performed over the public dataset Bar Crawl: Detecting Heavy Drinking Data Set. It contains acceleration recordings from 13 subjects during a university event and measurements from a TAC sensor. The results significantly outperform the performance reported in previous works over the same dataset.

This paper is organized as follows. Section 2 reviews the related work. Section 3 describes the material and methods, including the dataset, the signal processing and deep learning modules. Section 4 details the evaluation metrics and the experiments performed in this work. Finally, Section 5 summarizes the main conclusions of this work.

II. RELATED WORK

Alcohol detection through mobile sensing has gained popularity in the last years. Researchers have combined different sources of information from smartphones and wearable devices, such as acceleration signals, location, keystroke speed, or sent/received calls to predict intoxication levels of alcohol consumers. Moreover, inertial signals from wearables and smart devices have been used for motion modelling in other areas, like activity classification [1][2] or biometrics [3]. This section describes several previous works on alcohol detection based on mobile sensing.

Kao et al. [4] developed controlled laboratory experiments to classify alcohol intoxication through smartphone accelerometer signals. Arnold et al. [5] compared several machine learning algorithms (Naïve Bayes, Decision Tree, Support Vector Machines and Random Forest) using acceleration data from the smartphone to classify alcohol intoxication levels through the number of drinks consumed by a user. They proved that Random Forest was the most accurate classifier, reaching 56% and 70% accuracy on the training and validation sets, respectively, classifying the number of drinks into ranges of 0-2 drinks (sober), 3-6 drinks (tipsy) or >6 drinks (drunk). This work reached encouraging results, but they used potentially biased self-reports to measure ground-truth intoxication levels, which could limit the reliability of the results.

Santani et al. [6] characterized youth drinking behavior using smartphones involving 241 participants during a weekend night using a Random Forest classification algorithm to infer whether an individual consumed alcohol (over a threshold). This work also used self-reports on individual alcoholic drinks consumed on Friday and Saturday nights over a three-month period. They concluded that accelerometer data was the most informative single signal, reaching an accuracy of 75.8%.

McAfee et al. [7] used drunk busters goggles to distort vision and simulate the effects of alcohol consumption on the body and rate at four BAC levels [0.00-0.08), [0.08-0.15), [0.15-0.25), [0.25+). They used accelerometer and gyroscope features from smartphone, height, weight, and gender reached to classify 33 subjects into these BAC levels. This previous work used 5-second segments and reached 89.45%

 ISBN: 978-1-61208-970-6

of accuracy when detecting the BAC level using a decision tree classifier when using 99% as training data and 1% as testing data and 73.74% using a Random Forest algorithm when using a 10-fold cross-validation setup.

Killian et al. [8] measured accelerometer signals with a smartphone and TAC data during a drinking event in a non-intrusively way. This work used 10-second windows of acceleration recordings and the authors randomized the data using 75% for training and 25% for testing. They compared machine and deep learning algorithms, concluding that Random Forest approach outperformed the classification between sober (TAC < 0.08) and intoxicated (TAC >= 0.08) participants and with a 77.48% of accuracy. Another previous work [9] used the same dataset and performed both classification and regression task using a K-fold cross-validation strategy. They used a Convolutional Neural Network (CNN) architecture and obtained an accuracy of 80.43 ± 0.21 % using 2-second windows for the classification task and a MSE of alcohol content estimation of 0.001559 ± 0.000011 g/dl for the regression task.

In addition, a previous work [10] analyzed the effect performance saturation in activity recognition when increasing the analysis window size. They proposed several strategies to combine the information from consecutive sub-windows, obtaining significant improvements compared to directly using long windows. This paper combines several sub-windows at the end of a CNN architecture, obtaining significant improvements compared to previous works using the same dataset.

III. MATERIAL AND METHODS

This section describes the dataset used for the experiments, the signal processing, the CNN, and the post-processing module for combining the scores from sub-windows.

A. Dataset

We used the Bar Crawl: Detecting Heavy Drinking Data Set" [8]. It includes recordings from 13 undergraduate students in a drinking event. The dataset includes acceleration signals from a sensor embedded in smartphones sampled at 40 Hz and TAC measurements collected with an ankle bracelet. A TAC=0.08 g/dl was used as the level to discriminate between intoxicated participants (TAC >= 0.08) and sober participants (TAC < 0.08). Participants joined in drinking activities without any instruction. For the classification task, we considered two classes: intoxicated and sober participants. For the regression task, our target was to estimate the TAC. The total duration of the dataset is 77 hours approximately. The acceleration values mostly vary between -4 and 4g, and Table 1 summarizes the acceleration and TAC signals statistics.

B. Signal Processing

We divided the accelerometer signals into non-overlapped consecutive sub-windows using a Hanning function (other functions were evaluated like Hamming or Blackman without significant differences). In this paper, the system provided a consumption score per sub-window, and we integrated consecutive scores to evaluate longer periods.

TABLE I. ACCELERATION AND TAC SIGNALS STATISTICS

Signal	Units	Min	Mean	Max
X	g	-43.335	-0.009	39.23
Y	g	-33.475	0.001	27.311
Z	g	-49.023	0.056	42.313
TAC	g/dl	0	0.065	0.443

For each sub-window, we computed the Fast Fourier Transform (FFT). For example, in case of using 3-second sub-windows, we used 60 bins in the frequency domain per example as inputs to the CNN corresponding to the FFT magnitude from 0 to 20 Hz. As in previous works using this dataset, we only considered temporal windows whose estimated energy was higher than zero at 2 Hz (average human walking activity frequency). We used GNU Octave for the signal processing step (windowing and computing the FFT).

C. Deep Learning Architecture

We used a deep learning approach composed of a feature learning subnet and a classification subnet. Figure 1 represents the architecture that models and classifies participants between intoxicated (TAC >= 0.08) and sober (TAC < 0.08) using 3-second sub-windows. The first part of the structure learns features from the spectra using three convolutional layers and one intermediate max-pooling layer. The second part of architecture contains fully connected layers that classify the sub-windows as intoxicated or sober subjects. The last layer has one neuron and uses the sigmoid activation function, and the binary cross-entropy loss metric for classification problem. In case of regression problem, this last layer has a linear activation function and uses the mean squared error loss metric. In intermediate layers, ReLU is used as activation function to reduce the impact of gradient vanishing effect. Both tasks used the root-mean-square propagation optimizer [11], with learning rates of 0.001 and 0.00005 for classification and regression tasks, respectively. It was discovered that to achieve better results on the regression task, a lower learning rate was required. Before reporting testing results, the validation subset (10 % of training subset) was used to tune the number of epochs (10) and the batch size (200) of the architecture. Each This architecture, which uses 3-second sub-windows, has 137,665 parameters. We used Python distribution with Tensorflow and Keras libraries to create the deep neural network architecture.

D. Post-processing Decision Module

Combining the information along several consecutive sub-windows allows increasing the decision robustness, by

 ISBN: 978-1-61208-970-6

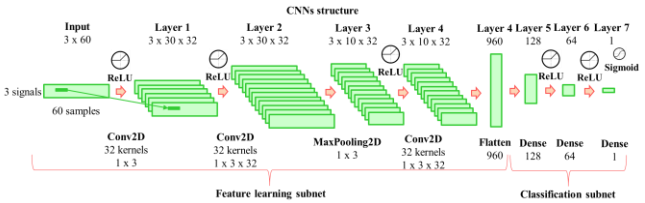

Figure 1. Deep learning architecture including convolutional and fully connected layers for classification. This is the deep learning module which obtains a score sub-window by sub-window.

evaluating long periods of time that keep a uniform behavior. The mean filtering approach used in this work consisted of running a non-overlapped filter through a specific signal and computing the mean of N sub-windows. In this sense, this mean filtering was used as a post-processing technique that allowed the integration of information from consecutive windows using the final scores of the CNN output. After computing the mean score of N consecutive windows, we obtained a single value which integrates the information along these windows. When the prediction is completely filtered, the final number of examples would be divided by N. We used N=1, 2, 4, 5, 6, 8, 10, 12, and 14, being N=1 the lack of filtering. Figure 2 shows an example of mean filtering of final prediction using N=4 with 3-second sub-windows, where the prediction (between 0 and 1) is modified after applying the filtering technique and integrating more time (12 s) and some isolated errors are corrected through this integration of temporal information.

IV. Experiments and Discussion

This section defines the evaluation metrics used in this work and shows the results in the experiments.

A. Evaluation Metrics and Validation

This paper performs two tasks: TAC classification and regression. For the classification task, we used accuracy: the ratio between the number of correctly classified examples and the number of total examples. In our case, every analysis window is considered as an example. This metric is presented with confidence intervals of 95%, obtained with (1), given M examples (windows) and a specific value of accuracy. Two results are considered significantly different when there is no overlap in these confidence intervals. We also used the Area Under the Curve (AUC) to evaluate this binary classification problem.

$$\text{acc}\ (95\%) = \text{acc} \pm 1.96 * \sqrt{((\text{acc}(100-\text{acc}))/M)} \quad (1)$$

Regarding regression task, Mean Square Error (MSE) was considered as the average squared difference between the estimated values and the actual values. This error is presented with confidence intervals of 95%, obtained with (2), given M examples (windows) and an error standard deviation s. We also used the Pearson correlation coefficient between the estimated and the actual TAC measurements to evaluate the regression problem.

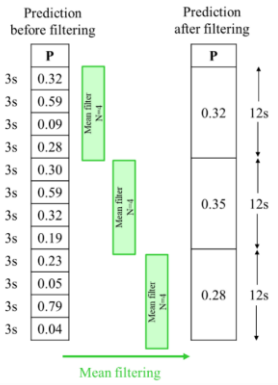

Figure 2. Mean filtering of predictions using N=4 and 3-second sub-windows.

$$\text{MSE}\ (95\%) = \text{MSE} \pm 1.96 * s/\sqrt{M} \quad (2)$$

In this work, we used K-fold cross-validation for comparison to previous works: data is divided into K folds (13 in this work) to divide data in training, validation, and testing subsets. A different fold is used for testing in each iteration, with the remaining folds used for training (10 % of training subset was used for validation). This methodology allows to evaluate the system over all available data using different data distributions. The reported results are the average along all iterations. For example, in case of using 3-second sub-windows, a total of 92,000 examples approximately are considered. For each fold, 7,000 examples for testing and 85,000 examples for training (8,500 for validation) approximately. Training the model with a 90% of data, guarantees a well-trained model: reducing this amount could have a negative impact over the performance. We observed these classification and regression tasks are high-user dependent, so a leave-one-out approach is considered as future work.

B. Experiments

We analyzed the influence of the window size and the combining information technique averaging the predictions from sub-windows after the deep learning architecture over the classification and regression tasks. As baseline system [9], we performed lack of combination experiments using long windows directly to observe the performance saturation when increasing the analysis window length. After that, we compared these results to our approach: averaging the scores from sub-windows after the CNN.

Related work section mentioned previous works [8] [9] that obtained 77.48 % and 80.43 ± 0.21 % of accuracy for the classification task and a MSE of alcohol content estimation of 0.001559 ± 0.000011 g/dl for the regression task.

Figure 3 and Figure 4 show the test accuracy and AUC, respectively, using the baseline approach (lack of combination), 3-second sub-windows, and 6-second sub-windows combination for the alcohol classification task. Figure 5 and Figure 6 show the test MSE and correlation,

 ISBN: 978-1-61208-970-6

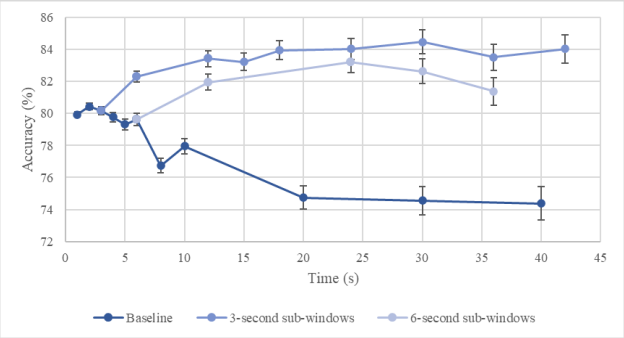

Figure 3. Accuracy evolution including the baseline results and the performance results obtained when integrating 3-second and 6-second sub-windows for the alcohol classification task.

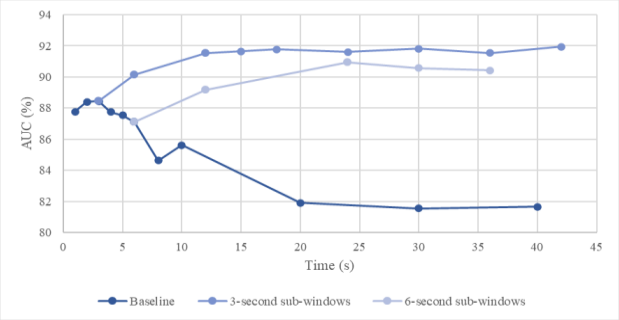

Figure 4. AUC evolution including the baseline results and the performance results obtained when integrating 3-second and 6-second sub-windows for the alcohol classification task.

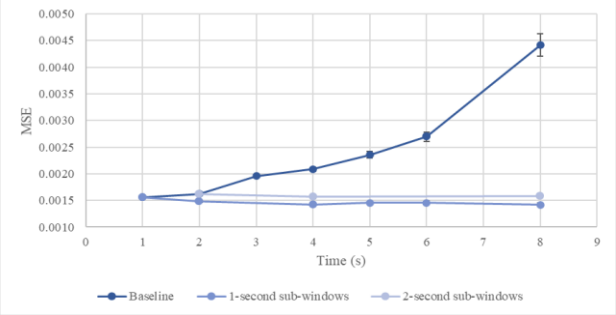

Figure 5. MSE evolution including the baseline results and the performance results obtained when integrating 1-second and 2-second sub-windows for the alcohol estimation task.

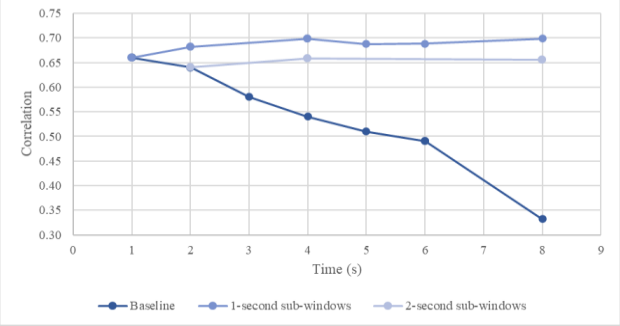

Figure 6. Correlation evolution including the baseline results and the performance results obtained when integrating 1-second and 2-second sub-windows for the alcohol estimation task.

respectively, using the baseline approach (lack of combination), 1-second sub-windows, and 2-second sub-windows combination for the alcohol estimation task. These figures show that the baseline approach achieves a performance saturation when increasing the analysis window length because we raise the number of parameters to be trained in the CNN and the spectral resolution, that increases the overfitting risk. However, integrating the scores after the CNN allows to boost the classification and regression performances, reaching a maximum in accuracy (84.47 ± 0.74 %) and AUC (91.82 %) evaluation metrics for 30 s evaluation using 3-second sub-windows. In the case of alcohol content estimation, the best result was an MSE of 0.00142 ± 0.00002 and a correlation of 0.7, obtained when combining four 1-second sub-windows.

V. Conclusion and Future Work

Detecting alcohol consumption through wearable technology and deep learning is very interesting to avoid health risks in the future. This paper contributes to the supervision of alcohol consumption from acceleration signals by proposing a method to evaluate long periods of time. The system leverages that alcohol content is quite stable in time to integrate information from short sub-windows and boost the classification and regression performances. Using these short sub-windows, it is possible to decrease the number of parameters to be trained in the CNN and reduce the overfitting risk that occurs when increasing the spectral resolution. This work used the Bar Crawl: Detecting Heavy Drinking Data Set, obtaining better performance than previous works that used the same dataset.

As future work, it would be interesting to leverage the sequential information from sub-windows using Long Short-Term Memory (LSTM) layers to analyze the evolution of the alcohol content. In addition, we observed that the current approach has the limitation of generalizing to unseen subjects, so it would be useful to apply adaptation techniques and focus on specific characteristics of subjects in a Leave-One-Subject-Out CV scenario.

Acknowledgment

The work leading to these results is part of the projects AMIC (MINECO, TIN2017-85854-C4-4-R), AMIC-PoC (PDC2021-120846-C42), CAVIAR (MINECO, TEC2017-84593-C2-1-R) and GOMINOLA (PID2020-118112RB-C22) funded by MINECO/AEI/10.13039/501100011033 and by "the European Union "NextGenerationEU/PRTR". We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research.

References

[1] M. Gil-Martin, R. San-Segundo, F. Fernandez-Martinez, and J. Ferreiros-Lopez, "Improving physical activity recognition using a new deep learning architecture and post-processing techniques," *Engineering Applications of Artificial Intelligence*, vol. 92, Jun 2020, pp. 103679, doi: 10.1016/j.engappai.2020.103679.

 ISBN: 978-1-61208-970-6

[2] M. Gil-Martín, R. San-Segundo, F. Fernández-Martínez, and R. de Córdoba, "Human activity recognition adapted to the type of movement," *Computers & Electrical Engineering*, vol. 88, pp. 106822, 2020/12/01/ 2020, doi: https://doi.org/10.1016/j.compeleceng.2020.106822.

[3] M. Gil-Martin, R. San-Segundo, R. de Cordoba, and J. Manuel Pardo, "Robust Biometrics from Motion Wearable Sensors Using a D-vector Approach," *Neural Processing Letters*, pp 2109-2125, 2020, doi: 10.1007/s11063-020-10339-z.

[4] H.-L. Kao, B.-J. Ho, A. C. Lin, H.-H. Chu, and M. Assoc Comp, "Phone-based Gait Analysis to Detect Alcohol Usage," *Ubicomp'12: Proceedings of the 2012 Acm International Conference on Ubiquitous Computing*, pp. 661-662, 2012 2012.

[5] Z. Arnold, D. LaRose, and E. Agu, "Smartphone Inference of Alcohol Consumption Levels from Gait," (in English), *2015 Ieee International Conference on Healthcare Informatics (Ichi 2015)*, Proceedings Paper pp. 417-426, 2015, doi: 10.1109/ichi.2015.59.

[6] D. Santani, T. M. T. Do, F. Labhart, S. Landolt, E. Kuntsche, and D. Gatica-Perez, "DrinkSense: Characterizing Youth Drinking Behavior Using Smartphones," (in English), *Ieee Transactions on Mobile Computing*, Article vol. 17, no. 10, pp. 2279-2292, Oct 2018, doi: 10.1109/tmc.2018.2797901.

[7] A. McAfee, J. Watson, B. Bianchi, C. Aiello, and E. Agu, "AlcoWear: Detecting Blood Alcohol Levels from Wearables," *2017 Ieee Smartworld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (Smartworld/Scalcom/Uic/Atc/Cbdcom/Iop/Sci)*, pp. 1-8, 2017, doi: 10.1109/UIC-ATC.2017.8397486.

[8] J. A. Killian, K. M. Passino, A. Nandi, D. R. Madden, and J. D. Clapp, "Learning to Detect Heavy Drinking Episodes Using Smartphone Accelerometer Data," in *KHD@IJCAI*, pp. 35-42, 2019. Available from: https://archive.ics.uci.edu/ml/datasets/Bar+Crawl%3A+Detecting+Heavy+Drinking, 2022.04.04

[9] M. Gil-Martin, R. San-Segundo, L. Fernando D'Haro, and J. Manuel Montero, "Robust Motion Biomarker for Alcohol Consumption," *Ieee Instrumentation & Measurement Magazine*, vol. 25, no. 1, pp. 83-87, Feb 2022, doi: 10.1109/mim.2022.9693446.

[10] M. Gil-Martin, R. San-Segundo, F. Fernandez-Martinez, and J. Ferreiros-Lopez, "Time Analysis in Human Activity Recognition," *Neural Processing Letters*, vol. 53, pp. 4507–4525 2021, doi: 10.1007/s11063-021-10611-w.

[11] N. A. Weiss, *Introductory statistics*. Pearson, 2017.

 ISBN: 978-1-61208-970-6