Instrumentation and Features Selection Using a Realistic Car Simulator in Order to Perform Efficient Single-User Drunkenness Analysis

Audrey ROBINEL
LAMIA,
Université Antilles-Guyane
Pointe-à-Pitre, France
audrey.robinel@univ-ag.fr

Didier PUZENAT
LAMIA,
Université Antilles-Guyane
Pointe-à-Pitre, France
didier.puzenat@univ-ag.fr

***Abstract*—We instrumented a car simulator by gathering low level data and fed it to an artificial neural network in order to perform blood alcohol content (BAC) estimations. The results depend on the quality of the data extraction and processing, and also on the selected inputs. We explain our data extraction and processing methodology, and how we used it to generate reliable and comparable features. At last, we describe the performances of individual features and how they combine. In the end, the prototype was able to accurately estimate the BAC value of a subject after being trained with driving samples of this subject with various BAC values.**

Keywords*-Blood Alcohol Content; Driving; Interface; Artificial Neural Networks; Intelligent systems; Machine learning; Instrumentation; Interface; Regression; Classification; Feature Selection; Car; Simulator;

I. INTRODUCTION

A. Problematic

Drunk driving is a major cause of accidents on the road. Alcohol alters people behaviour and ability to drive properly in many ways that are well known. We are working on a way to determine whether an user is in condition for driving or not, and estimate this user's blood alcohol content. The most common device for such a task is the breathalyzer. We however proposed a method for doing so without using a breathalyzer or any invasive device, by only monitoring how the user performs in ordinary tasks (the general methodology was described in [1]). In the current case, we monitor a car simulator controls and any available data from the embedded calculators, and analyze the driver's behaviour (we first presented the method for "car" monitoring in [2] with a racing game, and later with a car simulator in [3]). In order to do so, we had to train an artificial neural network (ANN) in order to make it capable of patterns detection within monitored data in real time. In this paper, we will discuss the methodology for data extraction and processing in order to perform this task. We will also discuss the issues that can impair the results, and present the monitored data. We will then describe some of the obtained features, along with results showing the performances of the network for each feature.

Figure 1. Our realistic car simulator (left) and its "force feedback" steering wheel with pedals (right).

B. Methodology summary

We will now provide a quick summary of the methodology (more on this in [1], [2] and [3]).
Our subject did consume an amount of alcohol, and then drove in our realistic car simulator ("Stars AF 2011", presented in [3] and provided by "ApportMédia") on a highway scenario with a countryside road at the end (Fig. 2). The simulator runs on a personal computer equipped with a Logitech G25 force feedback steering wheel with pedals (depicted in Fig. 1)

While the subject was driving, numerous low level data related to the controls and other parameters have been monitored. Before driving, the blood alcohol content value had been measured with a consumer class breathalyzer. We thereby obtained a set of monitored data associated with a BAC value. The subject repeated the operation (we call that a run) multiple times in order to provide examples with various BAC values so that we could create an example base.

Our final goal was to estimate the BAC value of the driver by feeding the monitored data to an ANN (a classical multilayer perceptron with back-propagation learning based

 ISBN: 978-1-61208-250-9

on the FANN library [5]). Whereas we could monitor many low level data, not all were suitable for our task. Furthermore, among the compatible data, we had to make some processing in order to be able to feed it to the ANN. We will at first describe how we gathered the low level data from the simulator. Thereafter, we will discuss how it had to be processed to produce features (potential ANN inputs), and afterwards present some of the obtained features. At last, we will present some results and discuss the impact of features selection on the ANN results.

C. On the realistic aspect of the simulator

When we use the term "realistic", we don't mean that the prototype used could replace a real car. Visually, we consider it realistic because it represents what an user would see in a car. It is not photo-realistic, but the cars are modelled after real cars, and so are the roads and environment. The software is considered realistic enough to be used for learning drivers in driving schools in France and Europe. However, in no mean we expect neither our prototype nor the software to replace real cars experimentation. Indeed, the user sees the virtual world through a screen that lacks the peripheral vision accessible in a car, and in this version, the controls are different from a real car controls (this point will howbeit be addressed in the next experiment, since we will be using real car controls in a hardware simulator with the same software). Furthermore, motion sensations are lacking, and various details too. Nevertheless, our prototype is realistic enough to analyze driving behaviour alterations caused by alcohol, and to validate our methodology and hypothesis.

Figure 2. The simulator software features "realistic" graphics.

II. Instrumentation of the simulator and monitoring of the controls.

A. low level data collection

The first task was to monitor low level data by instrumenting the simulator. Our simulator uses a script language that enabled us to read data from the physics engine (acceleration, speed ...) of the simulator, the position of objects (distance between cars, position on the road ...), and the state of all user controls (steering wheel, pedals, etc.). However, we can only read the instant value of those data at a given time. In order to be able to analyze the driver's behaviour, we have to be able to work on the evolution of the values of the variables. Since we had no way of knowing what data may or may not be useful before running the ANN, we decided to gather all the accessible values as often as possible. The simulator refreshes the values 50 times per second (one computation cycle per frame displayed on screen, and the frame rate is*fixed* at 50 per second), and all our variable can be read at each frame. We thereby obtain a vector of 28 values each 20ms. Gathering data at a constant time interval is important, since it enables us to compute the variation in time of each value easily.

B. collected data

An exhaustive list of the 28 variables instrumented is presented in Table I. The first column indicates the id of the variable, and the second gives a description of the value monitored. Although most are self-explanatory, we will detail those who are not. The "range" column indicates the interval of values than can be expected, completed by the type in the next column to describe the ensemble of possible values.

The variables 22 to 25 describe the surface under each wheel. In our scenario, it can be either 1 for road, or 2 for off-road. Other values are used for snow and other environments that was not present in our scenario.

We also gather the position of all vehicles, including the user's, but it is unused for now (other than for computing the distance to the closest vehicle in variable 14). Obviously, not all measures are meaningful in each experiment : here we decided to use an automatic gearbox car in the simulator, which explains why we don't use the clutch pedal readings.

C. Creation of suitable inputs from low level data

When the subject has finished driving on a scenario, we obtain a matrix of data that must be processed in order to be usable by the ANN. As we are not using a time based ANN model, we must generate suitable potential inputs (features) for the ANN. The simplest way to obtain a feature that reflect the behaviour of the subject on an interval of time is to use a statistical indicator, such as the average of the value. For some variable, such as vehicle speed, it provides a suitable feature whereas for other data (such as variable 20), it would not provide meaningful or comparable data . Therefore, we must for each variable study how it can be exploited and generate the corresponding features.

D. On normalization of features

Since the duration of the monitored run can vary, we had to ensure that the features used would be comparable. On the one hand, some features give the number of occurrences of an event, and must then be divided by the duration of the run. This provides us the frequency of the considered event, which does not depend on time (of course, in order to remain a significant parameter, the duration has to be long enough). On the other hand, other features were generated using averaged values, or similar computation. In this case, there is no need to divide the feature by time, as they are

 ISBN: 978-1-61208-250-9

Table I
LOW LEVEL MEASURES INSTRUMENTED FROM THE SIMULATOR

ID	Name	Range	Type	Notes
0	Accelerator pedal position	0;1	float	
1	Brake pedal position	0;1	float	
2	Clutch pedal position	0;1	float	unused in this experiment
3	Steering wheel position	-1;-1	float	
4	Vehicle speed	0;60	float	
5	Vehicle roll	-1;1	float	
6	Vehicle pitch	-1;1	float	
7	Vehicle engine RPM	0-8000	int	unused.
8	Instantaneous fuel consumption	0+	float	unused.
9	Instantaneous CO_2 production	0+	float	unused.
10	Instantaneous CO production	0+	float	unused.
11	Instantaneous HC production	0+	float	unused.
12	Instantaneous PM production	0+	float	unused.
13	Instantaneous NO_X production	0+	float	unused.
14	Distance to closest vehicle	0+	float	
15	Front wheel sleep angle	-1;1	float	
16	Rear wheels sleep angle	-1;1	float	
17	Vehicle acceleration on $\vec{X}$ axis	-1;1	float	
18	Vehicle acceleration on $\vec{Y}$ axis	-1;1	float	
19	Vehicle acceleration on $\vec{Z}$ axis	-1;1	float	
20	Engine gear ratio	{0,1,2,3,4,5,6}	int	0=no gear engaged, and 6=back. Unused.
21	Lights state	{0,1,2,3}	int	unused
22	Surface under front left wheel	{0,1,2,3,4,5}	int	1=road, 3=offroad
23	Surface under front right wheel	{0,1,2,3,4,5}	int	1=road, 3=offroad
24	Surface under rear left wheel	{0,1,2,3,4,5}	int	1=road, 3=offroad
25	Surface under rear right wheel	{0,1,2,3,4,5}	int	1=road, 3=offroad
26	Left indicator state	0 or 1	bool	unused
27	Right indicator state	0 or 1	bool	unused

already comparable. When needed, we used time as our divider. But we also could have used the number of measures used for the considered feature. Some features can have quite different ranges, so in the end, all values should be normalized relatively to each other to use the same range (in our case 0 to 1 or -1 to 1). We did so, and we often noted an improvement of experimental results when doing it.

E. How much pre-processing should be done on the data?

We made a design choice of not introducing human intelligence in the analysis of data (or as few as possible). We could indeed probably have simplified the problem for the ANN by constructing more complex, and higher level features that would be very specific to the problem of blood alcohol content. It may thus reduce the amount of work necessary for ANN optimization. However, doing so would also make our device dedicated to our current problem, whereas we wanted it to be usable for a wider class of problems related to drivers. We still have to provide results on other problems, but the designed prototype will be usable with no further modification.

Overall, we tried to create generic and and simple features in order avoid being specific.

III. PRECISION OF THE MEASURES

Precise measurement is important in order to create an example base. Indeed, the more precise are the measures, the less loss of accuracy we can expect on the trained ANN.

A. On instrumented variables

For the instrumented variables, in our case, we had no problem with the accuracy of the variables instrumented, as we use a simulated environment. The measures was therefore completely exact whereas when instrumenting a real device, the accuracy of the measurements must be taken into account. However, if we can have access to numerous variables, it becomes possible to select the most accurate for features generation. Furthermore, the combination of multiple features may compensate inaccurate measures.

B. Breathalyzer accuracy issues

The other part of the problem is the measurement of the BAC of the subject (or the expected output of the network, in a general case). We conducted tests on several subjects, and found that our breathalyzer can provide noisy measures, as shown in Fig. 3. In the first case, the BAC value increases of $+0.11g.l{-}1$ in 7 minutes before and after a decrease. In the second case, it decreases by $-0.09g.l{-}1$ in 5 minutes in

 ISBN: 978-1-61208-250-9

the second case. The decrease of BAC value in the second case is 6 times bigger than the average alcohol elimination value of human beings, and in the first case, the variation is quite unstable. We can conclude that the variations shown here depicts an important noise in measures, thus degrading our device accuracy. In the first case such an increase is not a problem, however we should have a constant variation, and not an increase preceded and followed by decreases, considering that the subject did not consume any alcohol for 20 minutes. For this subject, we think that an important part of the variation is noise. This is not the case for all subjects, but in this experiment, we could not have more reliable measures with a consumer class device. In the worst cases presented here, the noise could be up to $0.1g.l^{-1}$, which degrades the accuracy that we can obtain.

Ideally, we should have used a law enforcement class breathalyzer, but the costs for such a device is considerably higher than ours. Furthermore, obtaining such a device may be difficult due to the fact that they are not meant to be sold to regular citizens.

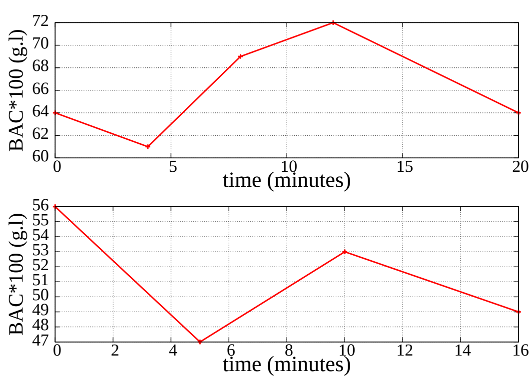

Figure 3. Evolution of BAC over time for two subjects.

When excluding some subjects that had excessive variations, we increased the accuracy of the prototype. However, we decided to keep as many subjects as possible in order to have more generalist results.

IV. Obtained features

After processing the low level data collected, we generated 25 features, that are presented in Table II.

For each feature obtained, we have an identifier (F0 to F24), and a descriptive name. Most are self explanatory, but we will describe those who are not. The third row indicates if this feature is already normalized or not. Then the "from" row tells what low level data from the Table I was used to generate the feature.

For the following rows, we present results that will be detailed later on. The idea is that we created a training base using only one feature, trained the network, and then tested it in generalization. We present regression results (R_x) that will be detailed in section V.

As mentioned earlier, we did not use the gear controls in this experiment. We however plan to use the clutch pedal and gearbox selector in future experiments. It must be noted that we can generate other features from the car's controls, but we decided to start with a few and then expand the list. We can thus consider the use of sub-features such as the average variation of a pedal when increasing the pressure, and when decreasing the pressure, or the duration of an action on the controls (average brake use duration, average duration of acceleration pedal increase time, decrease time...)

For now, we will explain some the used features. Many are related to the steering wheel, and steering wheel actions. We define a steering wheel action as a sequence of measures that starts when the steering wheel leaves the neutral (centre) position, and ends when it comes back to this neutral position. F0 gives the average duration of those actions. F6 is a variation in which we consider only the part of the action where the user turns the wheel in a constant direction. We stop the timer when he reverts the rotation of the wheel. F5 gives the count of direction changes operated on the steering wheel. F3 monitors how often the user was turning to one side before moving the steering wheel in the opposite side (e.g. the wheel was turned left, then reverts to neutral before going into the right half). For F1 and F4, we use "proximity alerts", which means that the distance to the closest vehicle goes below a fixed value. F1 gives the amount of measures where it occurred, while F4 counts the number of sequences where it happened (a sequence begins when the subject drives below the safety distance and ends when he goes farther. The set of measures is counted as *one* proximity alert sequence).

V. Results

A. Measurement of the ANN success rate in generalization

Using K-Fold cross validation, we test our network in generalization. When we perform classification tasks, if the output of the ANN corresponds to the expected class, we count a valid response, and otherwise we count an error. In the end, we divide the number of valid responses by the number of examples tested in generalization.

For regression purposes, we had to introduce a maximal tolerated error, ϵ. For each value returned by the ANN, we compute the distance between this value (N) and the expected value (E) : $dist = |E - N|$. If it is below a fixed epsilon, we count a success. Otherwise, we count a failure. We then compute the success rate of the ANN in generalization.

B. Analysis of the individual features results

The subject performed a total of 28 runs, which corresponds to approximately one hour and a half of driving in the simulator. Due to the fact that most of the examples had

Table II
FEATURES EXTRACTED FROM LOW LEVEL MEASURES.

The "source" row gives the corresponding measure used from table I. R1 gives individual success rates for $\epsilon = 0.2$, and R2 for $\epsilon = 0.1$. Both are expressed in percent. "Avg error" gives the average error of the network in $g.l^{-1}$.

ID	Name	Normalized	Source	R1:$\epsilon = 0.2$	R2:$\epsilon = 0.1$	Avg error
F0	Average steering wheel action duration	yes	3	75	50	0.126
F1	# of proximity alert measures	no	14	92	46	0.124
F2	# of steering wheel actions	no	3	92	53	0.108
F3	# of car direction changes	no	3	78	50	0.106
F4	# of proximity alert sequences	no	14	75	46	0.106
F5	# of changes of rotation direction of steering wheel	no	3	89	50	0.122
F6	Avg duration of constant direction steering wheel actions	yes	3	78	50	0.124
F7	# of measures with a wheel out of the road	no	22-25	78	53	0.108
F8	# of measures with any wheel out of the road	no	22-25	85	53	0.117
F9	Average vehicle speed	yes	4	85	53	0.118
F10	Average steering wheel shift from neutral pos.	yes	3	96	46	0.109
F11	Average roll of the car	yes	5	85	53	0.111
F12	Average pitch of the car	yes	6	82	50	0.112
F13	Front wheels avg sleep angle	yes	15	85	50	0.118
F14	Rear wheels avg sleep angle	yes	16	82	46	0.123
F15	Average accelerator pedal position	yes	0	85	46	0.119
F16	Average brake pedal position	yes	1	82	46	0.119
F17	Average steering wheel position	yes	2	82	42	0.113
F18	Average vehicle acceleration on $\vec{X}$ axis	yes	17	96	53	0.116
F19	Average vehicle acceleration on $\vec{Y}$ axis	yes	18	96	53	0.111
F20	Average vehicle acceleration on $\vec{Z}$ axis	yes	19	82	50	0.120
F21	Average accelerator pedal variation	yes	0	85	46	0.116
F22	Average brake pedal variation	yes	1	78	46	0.120
F23	Average steering wheel variation	yes	3	100	50	0.108
F24	Average vehicle speed variation	yes	4	50	50	0.121

BAC values below $0.5g.l^{-1}$, we could not create an unbiased classification base, and then only present regression results. In the R1 column, we present the features individual success rates for $\epsilon = 0.2$ and in R2 for $\epsilon = 0.1$. When looking at the results of R1 ($\epsilon = 0.2$), we note important variations, with a results ranging from 50% for F24 to 100% for F23. Some features seem to be much more discriminant than other. As one feature reaches up to 100% of success rate, there is no point in combining inputs. We therefore decreased ϵ to 0.1. In that case, individual results are much lower(42 to 53%), and little variation exists between most features. It is hard do discriminate the best features, and we can see that the best features in R1 are not always the best features in R2.

C. Combining features

Now we will combine features in order to see how the network performs. When using the 4 best features, according to the results of R1 (F10,F18,F19,F23), we obtain a 75% success rate, with an average error of 0.089764 for $\epsilon = 0.1$ (do note that R1 results are for $\epsilon = 0.2$). With only one feature, we obtained at best 53%, and combining features improved significantly the results.

We will now combine features according to R2. However, the results of the features are close, so we can have multiple configuration of 4 of the best features. When using F2,F7,F8,F9, we reach a 71.43% success rate and an average error of 0.091848. With another configuration, F2,F7,F11,F18, we reach 78.57% and an error of 0.080105. In both cases, we kept an ϵ of 0.1.

When we use the 4 best features according to the average error (F2, F3, F4, F23), we obtain a 78.57% success rate, and an average error of 0.077460. This configuration reaches the best success rate until now, and obtains the lowest average error.

We tried to combine various features, without considering if they are among the "best" or not, but rather by using features related to varied controls or data. A good combination was F2 F7 F13 and F18, with a 82% success rate and an average error of 0.085317. Using various sources provided to offer good results, so we tried with more inputs (F2 F8 F13 F18 F19 F22) and reached 85.71% with an average error of 0.088704.

VI. CONCLUSIONS

Our prototype showed that it could reliably estimate the BAC of a subject. We were able to obtain success rates up to 85% in generalization, when training the device with less than one hour and a half of driving. If integrated in a real car, it means that we could quickly gather data to create a learning base. Of course, for blood alcohol content estimation, there would be some practical complications for the creation of the base. For all that, the described methodology can be applied to many physiological parameters estimation.

When a simple and cheap solution is available, it may not be interesting to use our method. However, some physiological parameters of a subject are quite complex to acquire, requiring invasive, pricy or long procedures. In such cases, our method could prove to be quite useful : the cost of the device in both time and money only has to be spent once to create a large enough example base, and won't require invasive procedures when used in real life situation. Furthermore, on contrary to some methods that provides a measure with a delay, we can provide estimations in a short time, and continuously, so that the variation of the monitored parameters could be considered.

The main downside of this method is that it may require a long search for the most suitable features. The quality of the features used must also be ensured : whereas the ANN won't use bad features, including those makes the selection of good inputs longer. Creating the system can be quite complex, but it should provide a reliable, transparent, fast, non-invasive and economic way of estimating user parameters. Of course, the more precise are the measures, the more accurate the prototype should be. This implies that increasing the initial investment in both cost of used measurement hardware and number of examples monitored should provide an improved device while maintaining exploitation costs at constant level.

The selection of features can't be done by only using the individual success rates of features for a fixed ϵ. As a matter of fact, the best features changed from a ϵ to another. When combining the best features according to the individual average error, we obtained the lowest average error, but not the best success rate. It seems hardly feasible to obtain a simple metric to establish the performance of a feature, considering that individual scores may not reflect the potential effectiveness of combined features. A more complex solution could be to use something like the saliency of Optimal Brain Damage [6] : after having trained the network with all features, we could study how the suppression of an input cell corresponding to a feature impacts the network's performances. This value could provide a good metric.

VII. Perspectives

In this paper, we presented the results of a single user example base, but we will try to create multi-user bases in order to determine if the system could learn from multiple subjects so that it could estimate the BAC of an unknown subject. Although it was not the scope of this paper, it will be interesting to see how multi-user bases impacts the results presented here.

Our next goal will be to proceed with experimentation on tiredness and attention using the same software, but in an hardware simulator, with realistic car controls. Instrumenting will be done on the same basis, but with more available data (such as gear ratio, pedal...). The use of the simulator hardware should provide a driving experience closer to real cars, and enable us to collect more accurate data. We are also considering the use of data related to events rather than the average behaviour of the subject, like variation of various parameters when specific events occur (e.g. an accident, a dangerous situation, a change of the driving conditions...).

In the long term, we are looking forward to conduct similar experiments into real cars or trucks. However, we need to keep improving our prototype in the simulator in order reduce the number of experimentation needed in real cars, so that we can reduce the potential cost of development.

We are also looking forward to determine whether or not a subject is able to drive, and if not, what impairs his skills up to the point that he or she should not drive. This will be much more complex and require a collaboration with researchers in medical science. It should however be a good illustration of a complex to establish diagnostic about the subject that can't yet be done automatically.

Overall, we want to be able to detect various causes that alters the ability to drive of a subject.

Acknowledgment

This work has been funded by ApportMédia (`www.apportmedia.fr`), *la Région Guadeloupe* (`www.cr-guadeloupe.fr`), and *the European Social Fund* (`ec.europa.eu/esf`).

References

[1] D. Puzenat and I. Verlut, Behaviour analysis through games using artificial neural networks. In *proceedings of the Third International Conferences on Advances in Computer-Human Interactions* (ACHI 2010), pages 134-138, Sint Maarten (Netherlands).

[2] A. Robinel and D. Puzenat, Real time drunkenness analysis through games using artificial neural networks. In *proceedings of the Fourth International Conferences on Advances in Computer-Human Interactions* (ACHI 2011), pages 206-211, Gosier (France).

[3] A. Robinel and D. Puzenat, Real Time Drunkenness Analysis in a Realistic Car Simulation. In *ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0 pages 85-90. Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420. (ESANN2012)

[4] ApportMédia, "StarsAF2011" realistic car simulator product online documentation `www.simucar.com/logiciels/STARS_AF.html`.

[5] S. Nissen, Implementation of a Fast Artificial Neural Network library (FANN). Technical report, Department of Computer Science University of Copenhagen, October 2003.

[6] LeCun and Al., Optimal Brain Damage. In Advances in Neural Information Processing Systems (NIPS 1989), vol 2