Enhanced ALIVE Mind Controller and Machine Learning to Detect Drowsiness While Driving

Jihene Rezgui, Younes Kechout, Félix Jobin
Laboratoire **R**echerche **I**nformatique **M**aisonneuve (LRIMa)
Montreal, Canada
jrezgui@cmaisonneuve.qc.ca

Abstract – **Thousands of road accidents occur each year, one of the main causes being drowsiness at the wheel. Therefore, this paper proposes an enhanced scheme called ALIVE Mind to detect drowsy drivers and take action on their car using Machine Learning and EEG. For that matter, we built and designed a circuit board called AMC 2.0 that allows a simple EEG headset to read brain waves from a driver and send this data to a computer via Bluetooth. Those signals are then saved and analyzed by a deep neural network to find if the driver is drowsy or is about to sleep. To evaluate our solution, we conducted simulations and collected brain signals of a subject driver while on a driving simulator. With those values, we were able to train a model to detect whether the subject was drowsy or awake. Finally, when the system detects that the driver is in a fatigued state, it can take control of the car to park it in a safe place. Preliminary results show the effectiveness of the ALIVE Mind project and how it outperforms previous works in terms of minimizing computational needs and improving the prototype's convenience and suitability for car constructors.**

Keywords: ALIVE Mind, controlling physical vehicle, EEG, AMC, Machine Learning, IoT.

I. INTRODUCTION

The number of vehicles sold each year grows almost exponentially. At the same time, the number of road accidents is dramatically increasing. NHTSA estimates about 1,550 deaths, 71,000 injuries and $12.5 billions in economic losses are attributed to driver fatigue each year, just in the United States of America. However, we believe that there are many ways to reduce the number of road accidents. In recent years several systems have been developed and implemented to reduce the number of road accidents, such as backup cameras or blind spot detection systems, which can avoid certain collisions. There are other systems such as traction control systems or the electronic stability control that helps keeping control of the car in hazardous conditions. The problem is that none of these systems considers the most important cause of road accidents, namely the human factor. By using the driver's biometric data, it would be easier to predict and avoid the various road accidents caused by anger, fatigue, or the level of sobriety. Several studies focused on this path by training Machine Learning (ML) models with data about the driver, such as images or electroencephalographic (EEG) signals, to detect if he or she is in a drowsy state [1-6]. However, these works lack convenience and affordability to be considered by car constructors. For this reason, we built an easy-to-use system, called ALIVE Mind, that can detect drowsy drivers and acts upon this real-time detection. To do so, we improved the printed circuit board (PCB), designed in our previous work which was called AMC 1.0 [7], to collect brain waves and trained a ML model to determine whether the driver is drowsy or awake. As soon as we know that the driver is about to sleep, we can take action to help the driver, such as calling the police, turning on the lights, ringing an alarm to wake up the driver or even taking control of the car and bring the driver to a safe place. We think that this project could be helpful in real life situations and could contribute to reduce the amount of car accidents due to driver fatigue.

Our contributions in this paper can be summarized as follows: **(1)** We designed a custom PCB named ALIVE Mind Controller 2.0 (AMC 2.0) which was integrated in a small and easy-to-wear EEG headset; **(2)** We proposed a new way to efficiently and effortlessly collect brain waves data using Bluetooth technology; **(3)** We analysed the collected data to build multiple ML models that can run with low computational demand; **(4)** We ran several experiments to achieve high accuracy with a restrained dataset and **(5)** We controlled a new physical ALIVE car [8-9] based on the brain waves and the best ML model's outputs.

Section II gives a brief overview of related works and compares them to our ALIVE Mind project. Section III describes ALIVE Mind architecture and its different components. Section IV explains the data preprocessing steps and ML models. Section V shows the ML algorithms performances as well as an overall comparison with previous studies under several criteria. Finally, Section VI concludes the paper and provides potential paths for future work.

II. RELATED WORK

Several projects have been proposed to tackle driver drowsiness detection with ML techniques which are in wide use today. These studies can be roughly divided into two categories: **a)** Computer vision for driver fatigue detection and **b)** Deep Learning and EEG. In this section, we present the main characteristics of each category. Then, we highlight important differences between studies in that category and ours.

A. Drowsiness Detection with Computer Vision vs ALIVE Mind Project

The studies falling into this category chose the computer vision approach to detect drowsiness on a driver. Their system would generally involve a camera recording the face and eyes movements. The captured images would then be fed into a ML algorithm that would predict the state of the subject. Many studies only use computer vision [1-2], but some decided to combine image processing with EEG signals [3], leading to better accuracy results in general.

Based on these studies, the body language of the face transmits a lot of information about the drowsy state of a driver, meaning that a model able to take these images as inputs can achieve remarkable performances. It also helps to detect if the driver is distracted while driving, which is another dangerous risk. However, many factors can affect the recorded data in the driver's environment like brightness, quick movements, and the driver's distance from the camera, which can negatively impact the model's predictions. Because ALIVE Mind scheme is using EEG signals to achieve the same results, we did not have to consider such factors and thus could focus on reducing the computing cost of our algorithms while keeping an acceptable accuracy.

B. Deep Learning and EEG for Driver Fatigue Detection vs ALIVE Mind Project

Studies that fall into the second category are much closer to ALIVE Mind project, because they predict the state of the driver based on EEG signals recorded by a headset on the driver's head. This headset includes electrodes strategically placed on specific zones of the brain to capture the signals, which are then sent to a computer with wires to train a ML model. Some studies have reported nearly perfect results in terms of accuracy [4] by using a large and deep model, while others have shown great performances with smaller models [5-6].

The common aspect of most studies in this category is the complexity and expensiveness of their hardware. For example, the headset used in [4] is a complex system with more than 10 electrodes and can cost thousands of dollars, while also being bulky for the driver. With ALIVE Mind, we aimed to use a simple, homemade headset that cost us just about $50 and let the driver almost forget that he is wearing it because of its light weight. Of course, with only 2 electrodes, our headset is most likely less precise in its data acquisition, but the accuracy results show that the trained models can still reach acceptable performances.

III. ALIVE MIND SCHEME

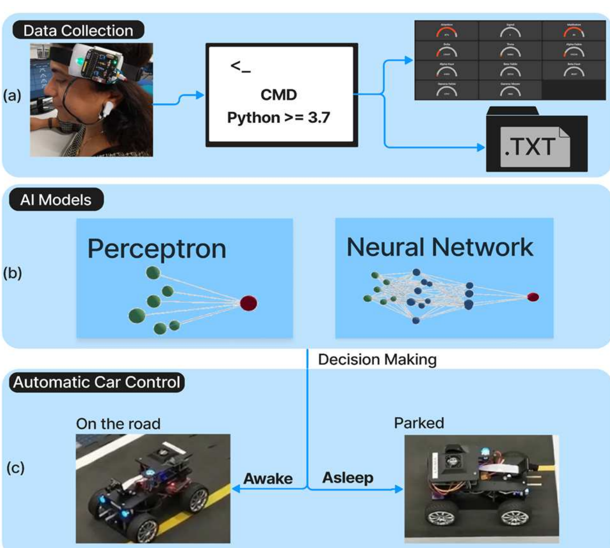


Fig.1. Architecture of ALIVE Mind

In its current state, the system architecture is made of three vital steps: (a) Data collection, (b) Communication and Data Analysis and (c) Automatic Car Control. The headset uses a TGAM module to transfer the data from the electrode to the AMC 2.0, the latter will organize and transmit the data by Bluetooth to a computer or a mobile device.

A. Data Collection

The headset transmits a variety of data, including signal strength and five different wave types, Delta, Theta, Alpha, Beta and Gamma. They represent the electrical diagram occurring in the brain of the headset user (see Fig.2). This headset is composed of two electrodes, one on the forehead for the collection of brain waves and another electrode on the earlobe as a reference. Even if the headset is composed only of two electrodes, it remains very advantageous. From its very low price (about 50$) as well as low energy consumption. Indeed, we can leave the headset for more than a week, to collect data.

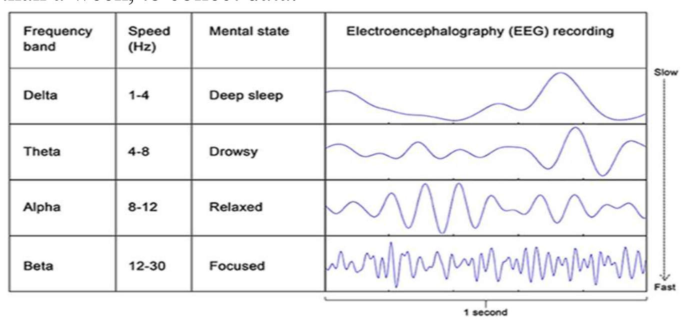


Fig.2. EEG waves characteristics

The microcontroller used in version 1 (see Fig.3.a) of the ALIVE Mind project has also been modified. The new version (see Fig.3.b) is up to 5 times smaller than the previous one but with more functionality. In addition to the basic components, we have replaced the batteries with a rechargeable 3.7-volt lithium-ion battery. A switch has been added to avoid draining the battery. A new green LED is added to indicate if the circuit is well powered.

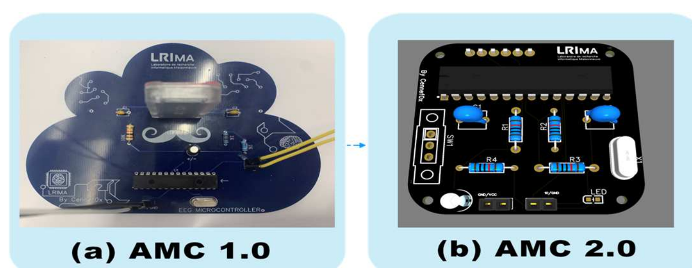


Fig.3. Side by side comparison between AMCs versions

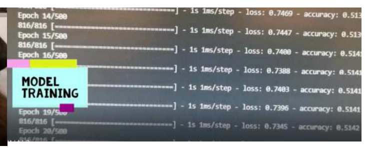
B. Communication and Data Analysis

Our headsets communicate via Bluetooth 2.0 with an HC-05 Bluetooth module. This module allows the headset to connect to Windows or Android, but not with Apple products, because they only accept modules using Bluetooth 4.0 or Bluetooth Low Energy. Electrode data is transmitted via Bluetooth with a script that we made in Python using a modified version of the PyBluez library. However, we had to use Python 3.7 because it is the most recent compatible version with this library. The task of this script is to detect the different Bluetooth devices around, obtain the MAC addresses of the different devices and establish a connection with the selected headset. Once the connection is established between the script and the headset, the data can be sent to the ALIVE server to later perform a broadcast of the data by Wi-Fi, print the data in the console or even save the data in an automatically generated text file.

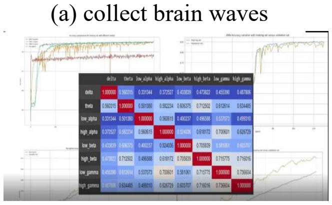
C. Automatic Car Control

We use a ALIVE car developed by LRIMa team to perform our simulations [8-9] (see Fig.4.(f)). This car has an autonomous driving algorithm that allows it to drive freely (see Fig4.(d)). However, we can regain control of the car at any time. It is possible to send commands to ask it to move forward, backward, turn the wheels to the right or to the left or to toggle the headlights. The data transmitted by the headset is treated and analysed by an ML models (Fig. 4 (b) and Fig. 4 (c)) to determine when to take control of the vehicle (Fig.4.(e)).

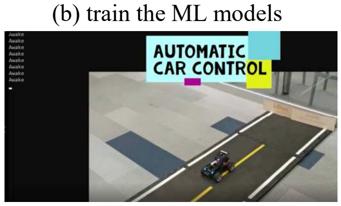
(a) collect brain waves



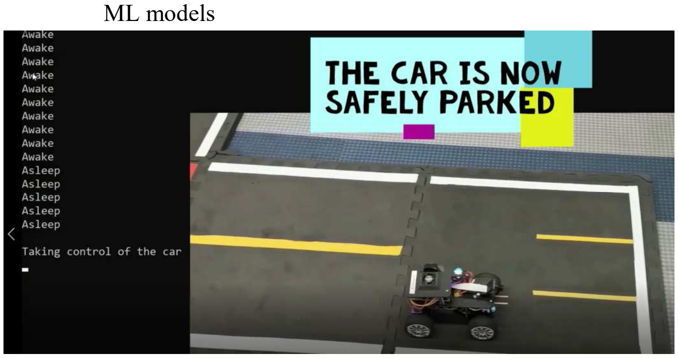
(b) train the ML models



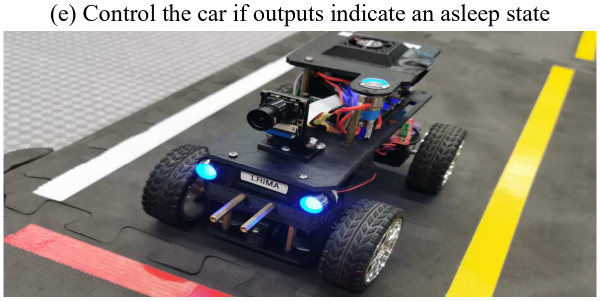
(c) Correlate data and compare ML models



(d) drive the car



(e) Control the car if outputs indicate an asleep state



(f) The self-driving car.

Fig. 4. Steps of drowsiness detection with ML models [10]

IV. MACHINE LEARNING ALGORITHMS

This section describes the two supervised ML model types analysed to achieve the drowsiness detection task, namely a perceptron and a deep neural network (DNN). But first, we begin with the preprocessing steps performed on the raw data.

A. The Preprocessing of Raw Data

Once the required data had been gathered by our AMC 2.0 headset as shown in section III, we began to analyse it to determine the necessary preprocessing steps to execute in order to get the best performances from our ML algorithms. Because the input data is only made of very high integers, we applied a min-max normalization on all input features to restrain all values between 0 and 1.

We also decided to remove possible correlations between input parameters. Previous research has shown that such correlations can significantly decrease a model's performance [11-12]. By removing highly correlated input features, we reduce the computing cost and improve the algorithm's accuracy at the same time. Therefore, we generated a correlation matrix that computes the correlation coefficient R for each pair of input features by using the following formula:

$$R = \frac{\sum_{i=1}^{n}[(x_i - \bar{x})(y_i - \bar{y})]}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \times \sum_{i=1}^{n}(y_i - \bar{y})^2}} \quad (1)$$

where $\bar{x}$ and $\bar{y}$ are the means of the two features for which we want to evaluate the correlation. The possible values of R range between -1 and 1, the former meaning a perfect inverse correlation and the latter meaning a perfect positive correlation. A value near 0 would describe a very small correlation with respect to the input features.

	delta	theta	low_alpha	high_alpha	low_beta	high_beta	low_gamma	high_gamma
delta	1.000000	0.560315	0.331344	0.372527	0.433839	0.473822	0.455390	0.487806
theta	0.560315	1.000000	0.501380	0.582234	0.606375	0.712502	0.612614	0.634485
low_alpha	0.331344	0.501380	1.000000	0.560615	0.400237	0.496588	0.537073	0.499310
high_alpha	0.372527	0.582234	0.560615	1.000000	0.524036	0.618172	0.700601	0.626729
low_beta	0.433839	0.606375	0.400237	0.524036	1.000000	0.705839	0.581061	0.603707
high_beta	0.473822	0.712502	0.496588	0.618172	0.705839	1.000000	0.715775	0.716016
low_gamma	0.455390	0.612614	0.537073	0.700601	0.581061	0.715775	1.000000	0.736604
high_gamma	0.487806	0.634485	0.499310	0.626729	0.603707	0.716016	0.736604	1.000000

Fig. 5. The correlation matrix of the brain waves captured by the headset's electrodes. A higher number signifies a stronger correlation between the 2 features.

As we can see in Figure 5, most of the coefficients are around 0.5, which is not significant enough to consider removing an input feature. However, most of the R values paired with low gamma and high beta are above 0.7. Because of these stronger correlations, the model's training could be affected by an over-representation of data relationships in which high beta and low gamma are a part of. Therefore, we decided to evaluate two sets of input features: one with all the features and the other without high beta and low gamma. In section V, we will compare the results between the two sets to see if this change brought any help to the model's training.

B. The Perceptron Model for the Brain Waves

The first ML model we considered to achieve the required task was a perceptron model. This type of model allows a large number of input features while keeping the complexity and the computing cost at a very low level. Our implementation of the perceptron model consists of an input layer of n features and an output layer of one Boolean value.

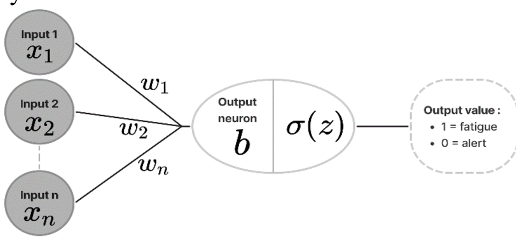


Fig. 6. Architecture of the perceptron model.

The output is generated by first computing the weighted sum of each input:

$$z = \sum_{i=1}^{n} w_i x_i + b \tag{2}$$

where n is the number of inputs, x_i the input value, w_i the associated weight and b the bias of the model. Both w_i and b are trainable parameters. We then apply the sigmoid activation function on the result:

$$y = \sigma(z) = \frac{1}{1 + e^z} \tag{3}$$

which compresses the values between 0 and 1. After that we round it to the nearest integer to obtain a binary output, where 1 stands for a tired state and 0 means an awake state. Figure 6 shows the perceptron's architecture with the location of its trainable parameters.

By considering this type of model, we aimed to find the minimum of complexity needed to achieve acceptable performances in driver fatigue detection. AI algorithms can be difficult to run on small devices because of their high computational demand while depending on less resources, thus motivating the need of finding a middle ground between accuracy and simplicity.

C. The Deep Neural Network Model for Brain Waves

The DNN was the second model type we considered in our work. A lot like the perceptron, many input features can be fed at the same time into this model. However, the addition of hidden layers allows models of this type to solve more complex situations, at the expense of more computational cost. A DNN computes its outputs by following the same kind of algorithm as the perceptron model but repeats it for each hidden layer in its architecture.

Because of the large number of weighted sums that must be done at the same time, it is common to use matrices in eq. 2, which results in the following equation:

$$Z_l = W_l X_l + B_l \tag{4}$$

where l is the index of the layer and the operator between W_l and X_l is a matrix multiplication. We also apply a non-linear activation function on the Matrix Z_l at each hidden layer, which is the Rectified Linear Unit (ReLU) function in our implementation:

$$X_{l+1} = ReLU(Z_l) = (0, Z_l) \tag{5}$$

where the function is applied to each element of the matrix Z_l. The resulting matrix X_{l+1} is then used as the input matrix of the next layer. For the output layer, however, eq. 3, namely the sigmoid function, is still used to reduce the output values between 0 and 1 as in the perceptron model.

Table 1. Search space and optimal value of some hyperparameters in the deep neural network.

Hyperparameter	Search Space	Optimal Value
Learning rate	$1 \times 10^{-3}, 1 \times 10^{-4}$	1×10^{-4}
Regularizer on each layer	None, L1 with $\lambda = \{1 \times 10^{-4}, 1 \times 10^{-5}\}$, L2 with $\lambda = \{1 \times 10^{-3}, 1 \times 10^{-4}\}$	L2 with $\lambda = 1 \times 10^{-3}$
Output activation function	Sigmoid, softmax	Sigmoid

We conducted several experimental training sessions in order to find the best set of hyperparameters. This set is shown in Table 1 while the resulting network architecture is displayed in Figure 7. Because of overfitting on the training set, we added a regularization technique on each layer of our optimal DNN. This addition helped to reduce some weights with high values that impact the model's ability to generalize on new data. This strategy has also been necessary in the perceptron model, as we will discuss in section V.

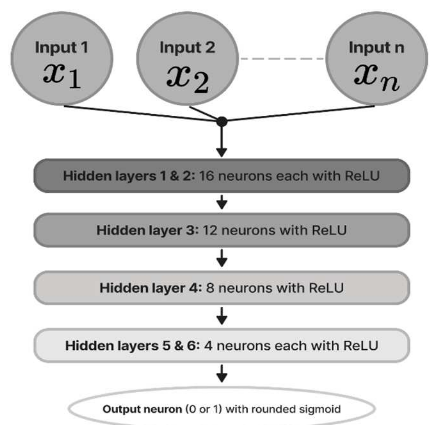


Fig. 7. Architecture of the proposed DNN.

V. RESULTS ANALYSIS

As experimental results, we first trained 4 slightly different models in order to find the best solution for driver fatigue detection. The parameters used in this simulation are described in subsection A, while the results are shown in subsection B. In subsection C, we compare ALIVE Mind scheme to other similar previous studies on many aspects.

A. Parameters of the Simulation

To train and compare our ML algorithms, we collected 11464 records of 8 brain waves from one subject using our ALIVE Mind headset. This data has been used to train 4 models of two different types in two input features arrangements. First, we evaluated the models with all of the collected brain waves as inputs. Second, we removed the two highest correlated input features, namely high beta, and low gamma, and evaluated all models with the six remaining input features.

Table 2. Differences in hyperparameters for the perceptron model and DNNs. The number in the name of DNNs denotes the number of hidden layers.

Model	Learning rate	# of neurons per layer	Regularizer on each layer
Perceptron	1×10^{-3}	-	L2 with $\lambda = 1 \times 10^{-4}$
DNN-4	1×10^{-4}	32, 24, 16, 8	L2 with $\lambda = 1 \times 10^{-3}$
DNN-5	1×10^{-4}	16, 12, 8, 4, 4	L2 with $\lambda = 1 \times 10^{-3}$
DNN-6	1×10^{-4}	16, 16, 12, 8, 4, 4	L2 with $\lambda = 1 \times 10^{-3}$

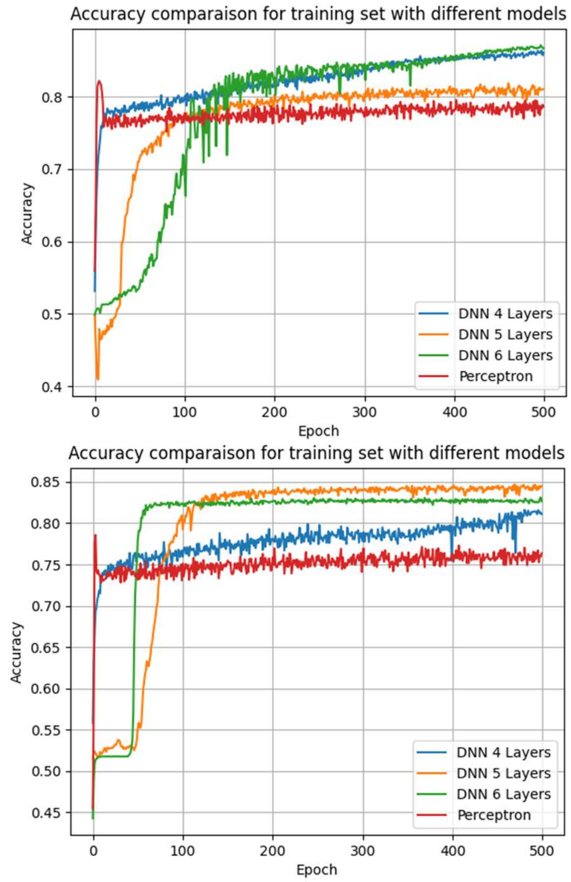
For the tested models, their main differences in their hyperparameters are shown in Table 2. The data has been randomly split by a ratio of 70/10/20 for training, validation, and testing sets. We also applied min-max normalization on each input feature to reduce the high values in the dataset.

Some hyperparameters are common for all tested models. Exactly like the model in Figure 8, all hidden layers of all DNNs have a ReLU activation function. Since the problem we are solving is a binary classification, we selected the binary cross-entropy loss function. Finally, we fixed the batch size at 10 and chose the Adam algorithm with $\beta_1 = 0.9, \beta_2 = 0.999$ and $\varepsilon = 1 \times 10^{-7}$ as the optimizer, because its implementation of gradient descent optimizes performance and computational cost [13].

B. Experimental Results

Based on the graphs shown in Figure 8, we can conclude that the results are quite different depending on the number of input features. With 8 inputs (Figure 8.a), the DNN with 6 layers is the most suited model to accomplish the task of detecting drowsiness. The higher number of layers have certainly helped it to generalize from the training data, hence the 0.86% difference between DNN-4 and DNN-6 as shown in Table 3. With 6 inputs (Figure 8.b), however, DNN-5 has the best performance while DNN-6 is second in training.

Overall, it seems like removing the most correlated brain waves have worsen the results, except for DNN-5. This may be caused by the fact that the input features were not correlated enough to negatively impact the training. Combined with the poor amount of available data, most models potentially needed the information contained in high beta and low gamma to improve their accuracy.



(a)

(b)

Fig. 8. Training results on the tested models. 8 inputs were used in 8.a while the most correlated features high beta and low gamma were removed in 8.b.

Nevertheless, these results show that a model as simple as a perceptron can achieve an accuracy over 80% for the task of driver fatigue detection, even when using the raw data from the AMC 2.0 headset. This overall simplicity can be compared with other similar studies on the subject.

Table 3. Accuracy results of the tested models with 8 and 6 input features.

Model	# of trainable params	Last training accuracy	Last validation accuracy	Test accuracy
Models with 8 input features				
Perceptron	9	78.63%	77.68%	85.14%
DNN-4	1625	86.11%	80.43%	85.14%
DNN-5	513	81.06%	73.05%	81.62%
DNN-6	**785**	**86.69%**	**83.26%**	**86.00%**
Models with 6 input features				
Perceptron	7	76.36%	80.60%	81.37%
DNN-4	1560	81.11%	81.46%	83.04%
DNN-5	**480**	**84.48%**	**84.46%**	**83.86%**
DNN-6	752	82.64%	81.20%	81.88%

C. ALIVE Mind compared to previous studies

Table 4 presents the main criteria of other similar studies in driver fatigue detection using EEG signals. These criteria have been selected to put in perspective the easier implementation of ALIVE Mind compared to others. If we want our solution to be effectively used in future cars to prevent drowsiness while driving, its implementation steps need to stay below a threshold that would discourage car constructors to try it on the field.

Table 4. ALIVE Mind project and previous studies compared on specific criteria. The number of * indicates in a range of 1 to 5 the degree of the criterion's appliance, while ' X' signifies the presence of the criterion and '–' means not present.

Criteria	ALIVE Mind	ANN [5]	Hybrid systems [3]	DNN [4]
Best accuracy (%)	86	83	94	99
Headset price (CAD)	$50	N/A	>$500	>$5000
Data acquisition convenience	*****	***	*	***
ML model complexity	**	*	*****	****
Restriction of driver movements	*	***	*****	****
Computer vision	-	-	X	-

As Table 4 shows, many other models can achieve better performance than ours. But their algorithms generally demand more computational power, because of deeper DNNs, convolutional or recurrent neural networks. Data acquisition, including data collection and data preprocessing, will also be more complex than what is needed in ALIVE Mind. Some will even need other sources of data to improve their model's accuracy as in [5].

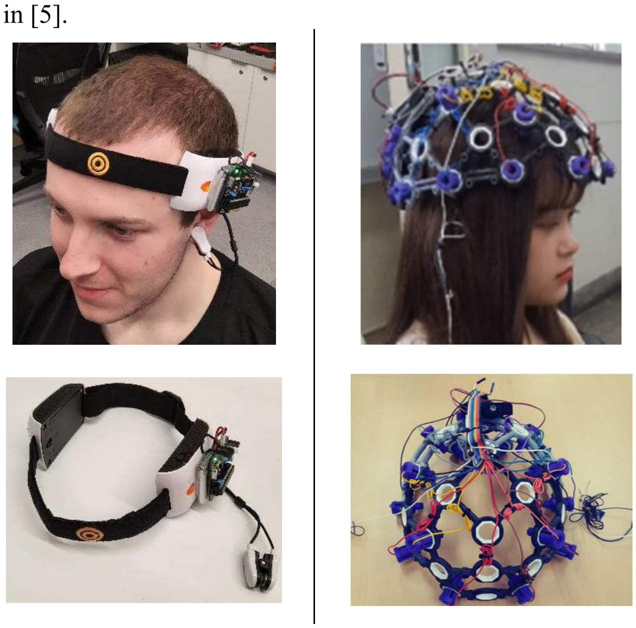


Fig. 9. Size comparison between ALIVE Mind Controller (AMC) 2.0 headset versus OpenBCI Mark IV headset in [3].

On the hardware side, our homemade headset is by far easier to wear and connect than headsets of other studies while being also more affordable. This statement is clear when we look at a visual comparison between the AMC 2.0 headset and the headset used in one of the studies in Figure 9. Finally, it is important to note that, because of the Bluetooth technology used in ALIVE Mind scheme, a driver with this headset would not be restrained in its movements while driving, while other wired headsets would force the driver to limit the movements of his head.

VI. CONCLUSION AND FUTURE WORK

In conclusion, the ALIVE Mind project has improved many components of the AMC to make it easier to use. Our goal was to find an efficient and easy way to collect brain waves data using Bluetooth technology and propose an adequate way to reduce road accidents caused by fatigue. We have proven that using ML with AMC 2.0 is an effective way to accomplish this goal.

For more improvements, we plan to collect more data with multiple subjects and to consider other types of ML models.

ACKNOWLEDGMENT

We gratefully thank the FRQNT and Mitacs for financial support. We also wish to thank other members of LRIMa Laboratory for their help throughout the course of this work.

REFERENCES

[1] RoseBrock, Adrian. (2017, May). Drowsiness Detection Using OpenCV. In *pyimagesearch*. pyimagesearch.com/2017/05/08/drowsiness-detection-opencv/ [last visited on August 2022].

[2] Jayanthi, D., & Bommy, M. (2012). Vision-based real-time driver fatigue detection system for efficient vehicle control. In *Int J Eng Adv Technol*, 2(1), 238-242.

[3] Karuppusamy, N. S., & Kang, B. Y. (2020). Multimodal system to detect driver fatigue using EEG, gyroscope, and image processing. In *IEEE Access, 8, 129645-129667*.

[4] Sheykhivand, S., Rezaii, T. Y., Meshgini, S., Makoui, S., & Farzamnia, A. (2022). Developing a Deep Neural Network for Driver Fatigue Detection Using EEG Signals Based on Compressed Sensing. In *Sustainability*, 14(5), 2941.

[5] King, L. M., Nguyen, H. T., & Lal, S. K. L. (2006, August). Early driver fatigue detection from electroencephalography signals using artificial neural networks. In *2006 International Conference of the IEEE Engineering in Medicine and Biology Society* (pp. 2187-2190). IEEE.

[6] King, L. M., Nguyen, H. T., & Lal, S. K. L. (2006, August). Early driver fatigue detection from electroencephalography signals using artificial neural networks. In *2006 International Conference of the IEEE Engineering in Medicine and Biology Society* (pp. 2187-2190). IEEE.

[7] Rezgui, J., Soldevila, E., & Kechout, Y. (2021). Novel Mind Controller to Assess Student Concentration with Connected Vehicles: ALIVE Mind. In *2021 International Symposium on Networks, Computers and Communications (ISNCC)* (pp. 1-6). IEEE.

[8] Rezgui, J., Gagné, É., & Blain, G. (2020, October). Autonomous Learning Intelligent Vehicles Engineering: ALIVE 1.0. In *2020 International Symposium on Networks, Computers and Communications (ISNCC)* (pp. 1-6). IEEE.

[9] Rezgui, J., Gagné, É., Blain, G., St-Pierre, O., & Harvey, M. (2020, October). Platooning of autonomous vehicles with artificial intelligence V2I communications and navigation algorithm. In *2020 Global Information Infrastructure and Networking Symposium (GIIS)* (pp. 1-6). IEEE.

[10] Simulation Video: [last visited 15 August 2022]. https://www.youtube.com/watch?v=4WA9VtUpug8

[11] Tang, J., Alelyani, S., & Liu, H. (2014). Feature selection for classification: A review. In *Data Classification: Algorithms and Applications (pp. 37-64)*. CRC Press.

[12] Sautter, J., Faubel, F., & Schmidt, G. (2018). Feature selection for DNN-based bandwidth extension. In *Proc. Jahrestagung für Akustik (DAGA)*, 43.

[13] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. In *arXiv preprint arXiv:1412.6980*.