

Advanced Learner Assistance System (ALAS) for engineering education using wearable sensors

J.M. Garcia-Briones
Tecnológico de Monterrey
Monterrey, México
A00825852@itesm.mx

G. Olivas-Martínez
Tecnológico de Monterrey
Monterrey, México
A01196915@itesm.mx

R. Abrego-Ramos
Tecnológico de Monterrey
Monterrey, México
A01720136@itesm.mx

M.O. Candela-Leal
Tecnológico de Monterrey
Monterrey, México
A01197730@itesm.mx

D.M. Botín-Sanabria
Tecnológico de Monterrey
Monterrey, México
A00825786@itesm.mx

G.J. Álvarez-Espinoza
Tecnológico de Monterrey
Monterrey, México
A00825719@itesm.mx

K.L. Rodríguez-Hernández
Tecnológico de Monterrey
Monterrey, México
A01197734@itesm.mx

J.F. Pacheco-Quintana
Tecnológico de Monterrey
Monterrey, México
A01373488@itesm.mx

S. Sampogna-Montemayor
Tecnológico de Monterrey
Monterrey, México
A01721245@itesm.mx

M.A. Ramirez-Moreno
Tecnológico de Monterrey
Monterrey, México
mauricio.ramirezmt@tec.mx

J.J. Lozoya-Santos
Tecnológico de Monterrey
Monterrey, México
jorge.lozoya@tec.mx

Abstract—This work presents neuroengineering tools used to assess students' cognitive performance under different learning modalities. By using a Raspberry Pi for IoT it is possible to use EEG helmets, other physiological signals, and vehicle signals to be analyzed using ML algorithms and give biofeedback on a web platform for augmented learning and make simulations for automotive engineering purposes. In a preliminary study, a correlation between cognitive performance and EEG analysis found the most relevant features to predict students' performance with an 85% precision at predicting and identifying the best learning modality (video > text). The current paper shows the first version of the functional system integrated, which includes three biometric devices and a predicted mental fatigue metric.

I. INTRODUCTION

A. General context

Education needs to adapt constantly, hence a variety of trustworthy evaluation and teaching methods need to be employed. A recent option is the use of biometrics as a biofeedback tool that works as an intelligent system for students, as well as teachers, to improve their respective learning performance.

Biometrics is the use of biological measurements related to the human body, one of the most oldest biometrics being used are fingerprints, since they have been used on forensics with the purpose of identifying an individual, as each fingerprint is unique [1]. On the other hand, recent methods extract biometrics, such as vein biometrics, further used on the financial industry to authenticate financial transactions [2].

Moving back to education, multiple efforts have been made to implement them, advances range from its basic form as biometric scanners such as fingerprints or face recognition to access facilities or as a payment method within the campus, done via companies as identiMetrics and IDconnect [3]. Other

studies use neuroimaging techniques such as functional magnetic resonance imaging (fMRI), that helps in the visualization of the current set of exercises a student is doing [4].

Both of the previously mentioned cases represent opposites sides regarding pricing and dependency of biometrics. The current study aims for a middle point, a variety of wearables device that can extract electroencephalogram (EEG), electrocardiogram (ECG), as well as many other signals. Those devices have an accessible price while still offering great resolution to reliably evaluate an individual.

B. Delimitation of the object of study

Further exploring education's field, the current study focuses in smart classes using Internet of Things (IoT) devices, which the purpose of gathering quantitative biofeedback that could be further used as a metric by both lecturer and student in hopes of creating a more cohesive, learning environment.

The aim is to create an integrated framework of biometric devices with the capability to improve performance of a student's learning during education. The development uses multiple biometric devices that gather biometrics from the student, in order to deliver an accurate prediction about the level of fatigue of the student wearing the wireless devices, which is further uploaded on the web application in order to track the performance of a student.

The system was tested with a student which used the integrated system while exposed to a video lecture used by Tec de Monterrey's Automotive Engineering subject. A future module will be integrated in the system, integrating signals coming from a vehicle, so students can get an interactive approach to learning the subject.

By using these biometric signals, the system in question was capable enough of providing the user feedback about their respective fatigue level.

C. Problem Statement

As it was previously stated, education needs to innovate in order to deliver a modern and fulfilling learning experience to its students. This demands new metrics that are highly related to the student's cognition when exposed to different teaching methods, which then leads to the creation of a database that could be further analyzed to deliver high quality lectures for the best learning experience by each student.

The previous scenario can be achieved via an integrated system that takes into account multiple biometrics in order to create advanced metrics, further used to evaluate teaching techniques and methods. This would provide an holistic perspective that uses reliable procedures to collect the corresponding data, as well as interpreting it using machine learning models. The collection of these data is then delivered in a easy-to-use platform for a teacher, in sought of giving perspicuity to the teacher in question of their respective students' state.

D. Justification

Studies have been made, using single biometric devices, in order to predict and gather a student's cognitive feature. EEG features are used to identify if a student is paying attention [5], combined with an audio signal to increase the levels of attention to high when they are low [6]. Also, thermal infrared imaging can draw the level of attention of students within a smart classroom [7].

These biometrics can then be used to solve student's learning-related issues, such as assessments loads, which play an important role when taking into account student's mental health, considering that negative emotions such as stress and anxiety reduce student's learning rate [8]. The inclusion of biometrics in the classroom can also create new learning ways, for example, direct feedback to students combined with an appropriate implementation of teaching methods, improve the learning pace and retention [9].

Moreover, a review in [10] compiled a set of studies that used Portable Electroencephalogram (PEEG) in a classroom, although it recognized that these devices are not as accurate as laboratory biometric devices, the authors expressed that PEEGs would certainly be an important research's area that may even revolutionize traditional teaching.

E. Theoretical framework

1) *Biometric signals*: Body signals can be gathered depending on the section of the body and the specific device being used. For example, brain signals are gathered by an electroencephalogram (EEG), heart signals by an electrocardiogram (ECG), body temperature by an infrared sensor, along with other biometric devices.

2) *Internet of things*: Internet of things (IoT) refers to a network in which multiple interconnected devices are exchanging information to each other. This is a latent subject, fundamental to smart housing and house automation, although, it can also be helpful when connecting a variety of biometrics devices to deliver an individual's full report.

3) *Machine learning*: A computational field that aims to train a computer without specifically pointing out the method it should follow, the computer then uses complex mathematics and algorithms in sought to find patterns about specific tags or classes (Supervised learning), or search for their own (Unsupervised learning).

4) *Interactive learning*: This concept refers to the use of a technological tool, within a classroom setting, that enhances participation and attention of the students. It is related to active learning, which is the preparation of activities and dynamics in the educational field, with the objective to increase student's participation when exposed to a subject that is being learnt, consequently boosting learning.

F. Objectives

As far as the authors know, currently there is not an integrated system which generates reliable, quantitative data as a metric of a student's performance. This is the objective of the project, to create an interactive application that uses student's biometrics to further create a report used by the teacher or the student itself, so to improve learning via using the right activities and assessments that are less fatiguing to a student in question.

The system has also opportunity to utilize the other biometric systems (Hexiwear and Empatica) in further experiments to add new features of interest to the interactive platform. As for the current project, only EEG signals provided by the OpenBCI were used, as these electroencephalographic signals are highly correlated with mental fatigue.

G. Hypothesis

Our hypothesis is that, as a integrated system with multiple biometric devices is developed, accurate predictions on learning-related metrics will further improve education's system, as the teacher would receive direct biofeedback from each students. Even the current mental fatigue metric, obtained via ML algorithms, is still helpful to directly compare teaching methods and materials.

II. PROPOSAL

A. Methodology

1) *Hardware Integration*: Done via a Raspberry Pi, which then connected to the multiple biometric devices in order to create a fully-integrated module that utilizes the devices. The equipment that were integrated were the OpenBCI EEG helmet, the Empatica E4 smart band and the Hexiwear smart watch. Access to a Geotab platform was also integrated, proving information about pre-recorded vehicle data.

First, the connection to the Hexiwear will be described, in order to completely acquire the information from the

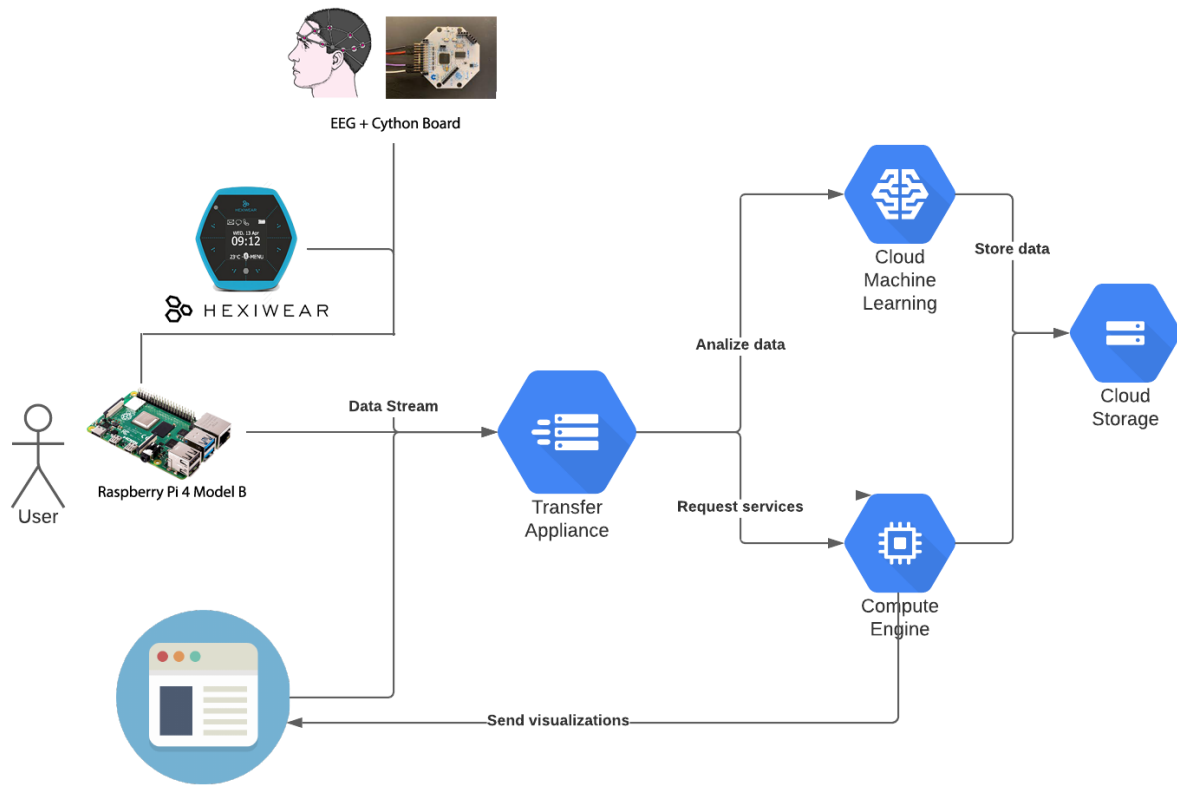


Fig. 1. Flow diagram of the overall integration within the multiple modules

Hexiwear and generate a CSV file in our Raspberry we utilized Professor Erich Styger's (From Lucerne University) tutorial from DZone (<https://dzone.com/articles/using-python-to-store-data-from-many-ble-devices>).

Said tutorial details step by step on how to establish a connection between both devices. Which is summarized on the following set of steps:

- 1) Installing BlueZ on raspbian
 - Package which enables Bluetooth Low Energy Features. Said feature permits the Raspberry Pi to connect with low energy Bluetooth devices, just like the Hexiwear.
- 2) Utilize "gatttool" to connect the Hexiwear and raspberry through Bluetooth.
 - Gatttool permits us to read sensor values or trigger notifications to be sent from the Hexiwear to the Raspberry Pi.
 - A full list of the Hexiwears sensors and the possible information which can be acquired from it can be observed in Fig. 5. (On our Hexiwear section)
- 3) Once the connection was established, and sensor values were able to be read, we utilized Python to utilize these tools and store the information from our sensors of choice in a comma separated values (CSV) file.

It is worth noting that the Hexiwear generates all values in hexadecimal digits. However, if a function is made in order to

convert said values into decimal, the values can be read and stored in our csv file without issue.

Raspberry was further connected to the Open BCI Cyton, which provided EEG signals, it was key that the Raspberry Pi runs python, as we were then able to utilize the code which we developed and explained in the OpenBCI section. However in order for the program to work we had to assure we completed two crucial steps correctly.

First, OpenBCI's Brainflow library had to be installed from source. This is because even though Brainflow provides a python build for Linux, it's only for x64 Linux architectures. In raspberry the arm is x86, which means that Brainflow's libraries should be compiled for its respective ARM. Brainflow's official documentation explains how (<https://brainflow.readthedocs.io/en/stable/BuildBrainFlow.html>). However it can be summarized in the following way:

- 1) We must install CMake
- 2) Download brainflow-master archive from Brainflow's official GitHub. Inside the archive we can find the project and the necessary compiler tools to build the library from source.
- 3) Run command to utilize the compiler tool and generate library locally.
- 4) Once compiled, access the Python package from terminal, then utilize PIP3 to compile to finally create the Brainflow library.

Once done, we must also assure that our Raspberry's usb port

is enabled and accessed accordingly. In our case, given that the Raspberry Pi 4 has 3.0USB ports, we took advantage of them and connected our Cyton dongle into that port. Further, we utilized the Raspberry's predetermined command known as "lsusb". This permitted us to identify our device ID. Furthermore, we utilized "udevadm" to know our port path name, and adjusted our Brainflow program accordingly using the devices ID and port name / path to access the Cyton board and stream the EEG information into the Raspberry Pi.

When working, we were able to utilize the Cyton board and our EEG to stream the information into the Raspberry and store all the data in a CSV file.

Regarding Machine Learning (ML), our algorithm functionality, theory and application is explained in its corresponding section, although it was also integrated via the Raspberry Pi.

In order for the Raspberry Pi to utilize our ML algorithm, we had to train it locally in the device. Meaning that we could not import it directly from a PC device given that most PC's run in 64-bit architecture, and thus train the model to utilize said computing system. The Raspberry, however, consists of a 32-bit architecture and can not utilize some imported algorithms. However, as stated: we fixed this issue by training the model locally. For more information on the Machine Learning section.

Moving on to tasks regarding data treatment, such as data posting, data storage, and our web software is explained in-depth in its corresponding section. However, it's worth mentioning some Raspberries functionalities which we had to make sure were done correctly.

First, all CSV files of relevancy, were converted into JSON formats utilizing Python as well. This was done with the purpose of making our entire platform as efficient and fast as possible. Additionally, the Raspberry utilized the application programming interface (API) our software development team did in order to post the JSON's into the server, for data storage and posting.

The posted information was the following:

- Hexiwear: Time, Temperature, and Heart rate.
- Brainflow: Generated PSD, and relevant wave files for investigation purposes.
- Machine Learning Algorithm: Fatigue analysis file.

2) *Machine learning*: The data used by the current study was obtained during a past study, it has fatigue classifications according to a questionnaire answered by the subjects before an activity, although posterior on the activity, the dataset lacks of fatigue classification. The data used was EEG data, extracted via a 8-channel OpenBCI filtered in order to get Alpha, Beta, Gamma, Delta and Theta signals, α , β , γ , δ and θ , respectively. These features were then normalized according to the mean (μ) of each feature, using calibration data which was gathered recording 30 seconds of eyes closed (EC) and eyes open (EO) by the subject, then using the Equation 1.

$$x_{norm} = \frac{x - \mu}{\mu} \quad (1)$$

Then, a set of combined features such as the ratio of signals within the same channel, were created by dividing each feature. For example, the ratio of δ and γ in the P7 channel of the OpenBCI is represented in the Equation 2.

$$\left(\frac{\delta}{\gamma}\right)_{P7} = \frac{\delta_{P7}}{\gamma_{P7}} \quad (2)$$

Afterward, as pre-processing, a filter equal to 10 was set, so that every value of each feature was between -10 and 10, and so $-10 \leq x \leq 10$. The total number of EEG features was 120, and so a A hybrid feature selection method was used, in sought to reduce to 10 features that are most related to fatigue, and so making it computationally feasible to predict a human's mental fatigue level in real time.

Firstly, the dataset with fatigue classes was fitted into a random forest classification algorithm to compute the Gini importance of each feature, which in essence is the information gain of the different classes according to each feature's node splits. Secondly, the top 20 features were selected and used as a filter to the whole dataset, we then used that dataset and fitted a Multivariate Adaptive Regression Splines (MARS) model to capture non-linear relationships between the features and so selected the top 10 features with the most number of subsets, and least residual sum of squares (RSS).

Lastly, the first pre-processed dataset was filtered with the top 10 features gathered and their values set to between -10 and 10, a final core machine learning classification model was further fitted on the dataset in order to predict an individual's fatigue level based on their EEG signals.

3) *Web development*: A web application based on microservices were developed for this project, the architecture of the whole application was divided between the backend and the frontend, the backend was developed in Django which is the most popular web framework for python, the efforts of the backend were focused on making a representational state transfer (REST) API for manage all the request needed by the users and for simplicity for the whole system as long as this project is based on an IoT Architecture this was the most optimal solution. Also, the backend is hosted on Heroku as a service. The frontend module was developed in Angular, a frontend framework developed by Google, as long as we wanted to make a simple and elegant User Interface (UI) we used several UI components from Nebular, and finally, the frontend is hosted on Google Firebase.

B. Methodological proposal to use

Our methodological proposal is represented on the Fig. 1, the workflow starts with the user in question that has interaction with the devices that provide the individual's biometrics, these devices are connected to a single Raspberry Pi 4 Model B, which is further connected to a web server and so the data is processed and sent to our generated ML model, as well as to our computer engine, to send the processed data back to the web application so the user can visualize it.

Meanwhile, the IoT data is stored in cloud storage so that the user can further visualize their past biometrics' recordings

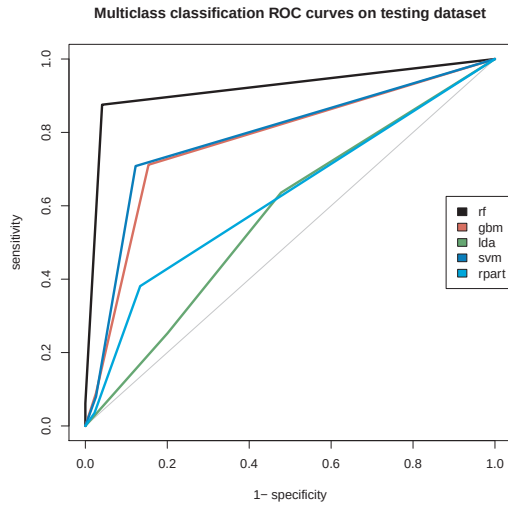


Fig. 2. Machine learning models trained and tested using the same dataset. Random Forest (rf) was superior, following Radial Support Vector Machine (svm) and Generalized Boosted Modeling (gbm), lastly, Recursive Partitioning and Regression Trees (rpart) and Linear Discriminant Analysis (lda).

and their respective levels of fatigue. This was also combined with Geotab and Google Maps to further implement the system in a real situation when the student learns about a topic, although the flexibility of the system lets it include additional biometric devices, as well as modules to improve learning.

C. Techniques and technological tools used

1) *Signal processing*: Electroencephalograph (EEG) signals on different frequency bands are related to emotional states, these were acquired via OpenBCI placing dry electrodes on the head of the subject, afterwards, a bandpass filter was applied to the received signals. Then, the power of each signal was calculated using the Fast Fourier Transform (FTT) in order to develop spectral analysis and classify each signal regarding the following ranges of values: δ (1–4 Hz), θ (4–7 Hz), α (8–12Hz), β (13–29Hz) and γ (30–50Hz).

2) *Machine learning*: Using the pre-processed and filtered data according to the most significant features, as well as the individual values of each feature. A Up sampling method was done on the whole dataset to reduce class imbalance, as "Moderate Fatigue" was a dominant class. Subsequently, a 80% split between training and testing data was done. A variety of core ML classification algorithms were further fitted into the train dataset using R and Python, each model's performance was then evaluated according to the accuracy of both training and testing datasets, as well as the sensitivity and specificity on each of the three classes.

The results of these parameters were summarized via Receiver Operating Characteristic (ROC) curves seen in the Fig. 2 in which the area under the curve (AUC) of each model represents the accuracy of the model. The comparison on each model's accuracy on the testing set can be further seen via the barplots presented on the Fig. 3, in which the random forest method performed best.

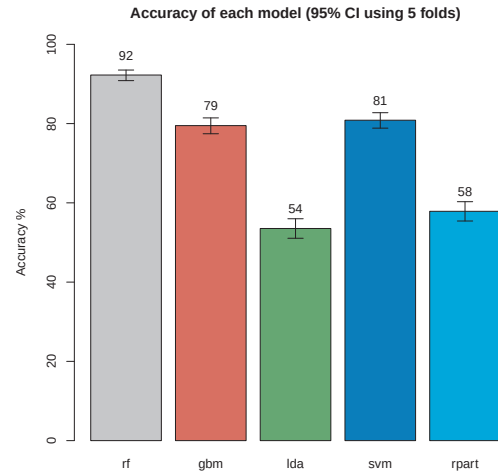


Fig. 3. A controlled set of conditions was applied for each model, Random Forest (rf) performed best with 92% accuracy. Multiple seeds were tested in order to acquire the error bars, accuracy was measured on the overall confusion matrix's accuracy score.

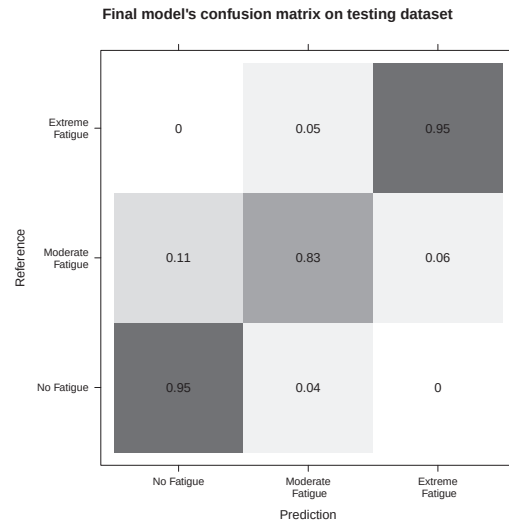


Fig. 4. Performance of the model on testing dataset expressed as a confusion matrix the model predicts accurately the extreme classes: "No Fatigue" and "Extreme Fatigue" (95% accuracy), although the most prevalent class 'Moderate Fatigue' predicted worse (83% accuracy)

A grid search was further performed in order to search for the parameters that maximized accuracy on the testing dataset and minimized RSS. Using these parameters, a final model was fitted on the training data set and the confusion matrix is presented on the Fig. 4.

3) *Software engineering*: As a software engineering project, it was a must to implement some software engineering techniques. First of all, for the management, the project used Agile methodologies, in this case, was Scrum with 2 weeks for each sprint. For collaboration, Git was the version control system we have chosen, as long as there are many modules in the project we opted to divide them, for the repositories that were attached to the frontend or the backend directly to the cloud providers we used CI/CD methodologies for

deployment. For the testing of the system, we used Selenium for the automated unit testing

D. Infrastructure

A web application which interconnects the results of multiple biometric devices (OpenBCI, Hexiwear and Empatica) on a single Firebase App, used via Geotab that uses Google Maps in order to provide an interactive UI to the user in question. Biometrics are injected into a ML algorithm to predict the level of mental fatigue that the user has during a session, and uploads it to cloud storage in order to inspect and contrast the performance of a student during multiple sessions using the ALAS infrastructure.

E. Resources used

1) *OpenBCI*: The Ultracortex “Mark IV” EEG headset is an open-source brain-computer helmet device that includes sensors, these sensors record the electrical activity of the body and collect the data at real time through different libraries in different programming languages, created by the company Open BCI.

To the acquisition of the information to process the EEG signals that the headset can give it to us, we use the programming language “Python”, and the library “Brainflow” to get the access of the data through the dongle included with the headset. This dongle is connected in one of the inputs of the raspberry and indicated on our code to identify where the data is provided. Also, with the library “Brainflow” we clean the signals that are being received and processed per second from the headset and create graphs of each brain wave and the data needed for this system with the library “Matplotlib”.

This part of the ALAS system generates different CSV files that bring the information for the machine learning module of the specific brain waves and the specific sensor of the headset to have the correct data to the predictive model.

2) *Hexiwear*: The Hexiwear is an internet of things smart device designed by the company MikroElektronika to serve as a multipurpose device. In this project it served two purposes. One utilizing its integrated HTU21D temperature sensor to constantly monitor the user’s temperature. Secondly, the MAX30101 integrated sensor was also utilized to analyse the users heart rate and any fluctuations through time.

It’s worth noting that the Hexiwear also possesses sensors which permit it to also act as an accelerometer, magnetometer, gyroscope, barometer, ambient light. Showing that the device has promise in order to utilize it in future projects, available on-board sensors are shown on Fig. 5.

Acquiring the data from this device and process is more challenging than that of the Empatica E4. Since this wearable has no server or intermediate application that can be connected directly to any programming language. Therefore, the team came up with the idea, based on literature, of using a Raspberry Pi 4 model B, configured with a Linux distribution (Raspbian OS) in order to establish a connection with the Bluetooth dedicated socket on-board the Hexiwear device. Having the UUID’s from each of the Hexiwear sensors, we

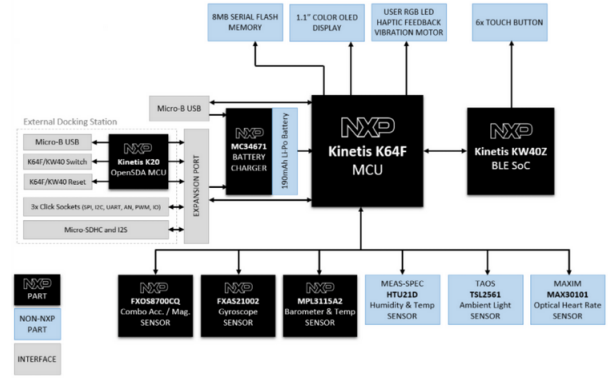


Fig. 5. Heiwear’s on-board sensors

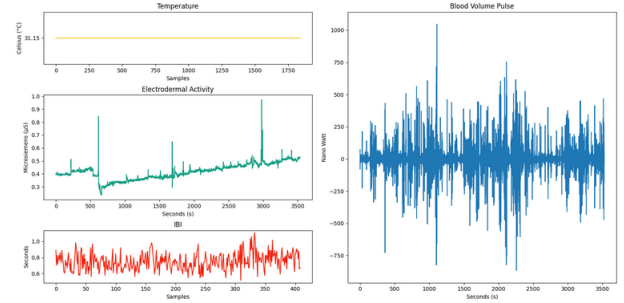


Fig. 6. Empatica E4 signal retrieval

were able to acquire the data using the Linux terminal and utilized python to access and store the data in real time.

3) *Empatica E4*: The E4 smartband is a device manufactured by the company Empatica and it provides us with a variety of signals such as an accelerometer, blood volume pulse (BVP), electrodermal activity (EDA) and the skin temperature.

With the BVP we are able to obtain the heart rate and heart rate variability, which in turn can be used to interpret if a subject is under stress, fatigue or in a relaxation period. EDA on the other hand can be used as a double-check of stress and fatigue since they can be derived from this signal too and cross-checked with the BVP results.

To acquire the information from this smartdevice, we took advantage of the python programming language and its ability to be connected with the Lab Stream Layer module. The LSL in turn, is able to interact with the “E4 streaming server” app available on Windows and MacOS platforms. This application is provided directly in the E4 Empatica web page. The E4 streaming server acquires the data from the smartband, and since python is able to communicate with the app, we can process this data. The system acquisition frequency is of approximately 1 Hz. With this system we are also able to monitor real-time data, an example is displayed on Fig. 6.

4) *Raspberry Pi*: The Raspberry Pi is the world’s third-best-selling general-purpose computer, after personal computers (PCs) and Macs [11]. It basically is a small computer running a Linux based operating system called Raspbian OS. It’s a truly versatile tool since it’s portable, cost effective, and

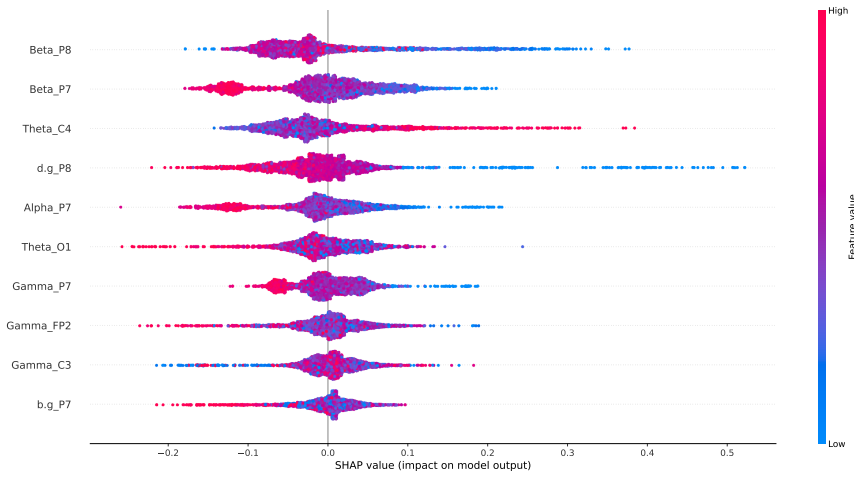


Fig. 7. SHAPley Additive exPlanations (SHAP) plot that explains the correlation of each feature to the predicted class

quite powerful. In our specific case, we utilized the Raspberry Pi 4 model B. Said device has a stunning 8 GB's of RAM, a Broadcom BCM2711 Quad core Cortex-A72 (ARM v8) 64-bit SoC @1.5GHz processor, HDMI ports, two USB 3.0 ports, other two USB 2.0 ports, Bluetooth connection, among other impressive specs for its small portable size.

The Raspberry Pi worked as the central node of our project. It was set up to run the Raspbian OS operating system. Once done, we utilized python libraries to activate the raspberries Bluetooth connection for it to connect with the Hexiwear. Furthermore, at one of the Raspberry's USB 3.0 ports we connected OpenBCI's USB Dongle to stream the electroencephalogram's data. Additionally, the Raspberry also ran a locally trained machine learning (ML) algorithm which incorporated the streamed information from the Cyton board and was able to determine the users fatigue levels through a classification method.

Once determined, and also having the Hexiwear's data, the Raspberry posted the information in a Heroku server which was interconnected with our web software that generated relevant graphs of our information in real time.

In the following section we detail specifically how the Raspberry Pi 4 was set-up and configured to connect with the devices and generate the ML algorithm. It's worth reaffirming that everything was programmed utilizing Python 3.8.

5) *Geotab*: A telematics provider which connects vehicles and provide web-based analytics. In order to get the images of a car driving across a street, data was extracted via an Application Programming Interfaces (API) connection to Geotab's Software Development Kit (SDK). Which in fact required to validate the connection, though it was accessible and useful when authenticating and a posterior use of its information.

The data was obtained setting up multiple devices in buses. This data required the *LogRecord* function, which required the vehicle's ID and the timeframe desired, the function then returns position and velocity of the vehicle's ID according to the date and time specified. Moreover, we have also used the *FuelUpEvent* function which returns fuel's consumption data.

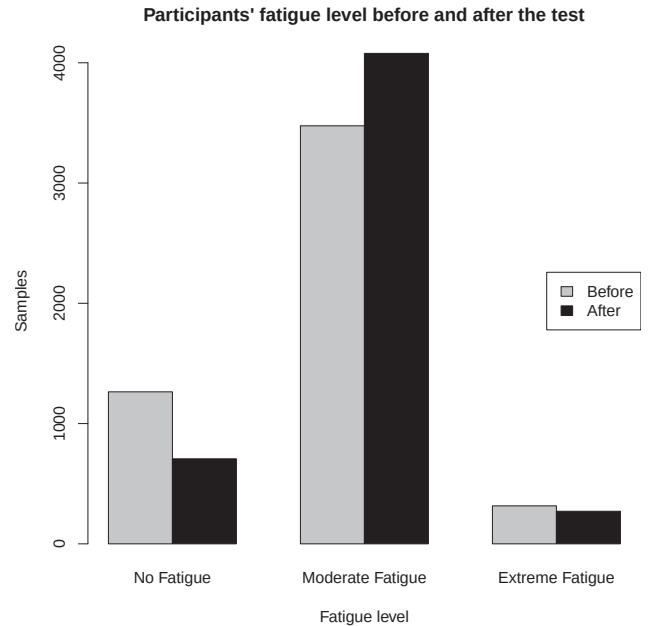


Fig. 8. EEG pre-test data contained specific fatigue classifications given by a questionnaire answered by participants, a section of this data was used to train the ML model. Afterwards, the model was used to predict fatigue level on the post-test EEG data, which had no true classifications given by participants. It was then found that there was a reduction of the "No Fatigue" classifications, similarly, the number of classifications of "Moderate Fatigue" incremented.

The aforementioned data is then saved as a JSON file, for space and speed efficiency purposes, and sent to our server. With the purpose of employing and displaying this information on the front-end, in addition to register it as a new entry.

Furthermore, using latitude and longitude data provided by the device, an image of the vehicle's location was created using Google Maps' API. Which was further integrated with the whole system in order to identify the fatigue level of a person which drives at specific speed and fuel's consumption values.

III. RESULTS & CONCLUSIONS

A. Results

A fully-functional integrated system was successfully created, which can be further improved with the addition of additional biometrics in order to predict and gather new features, further correlated with students' learning rate and attention. In addition, a single ML model was used to predict the mental fatigue of an individual. Further experiments, can then achieve multiple predictions in sought to find significant metrics that are hard to find using traditional, non-technological, techniques.

In regard of machine learning, the final selected model was then used on the post-dataset, in which only subjects' EEG signals were gathered, there was not a true classification of fatigue by the subject. And so the model predicted the fatigue classes of those signals, the results are displayed on the Fig. 8, where it seems clear that there was a reduction on the number of samples that were categorized as "No Fatigue", which is then related to an increase of the number of samples that presented "Moderate Fatigue", while the third classification, "Extreme Fatigue" remained similar before and after the test.

Although this results refer to data obtained by a previous study, it gives a certain degree of validation to our model, as it was successfully capable of generalizing on a completely new dataset. This was due to the fact that the increase of overall fatigue shown on Fig. 8 is explained by students completing a task, and thus were expected to experience more mental fatigue as compared to their baseline, and so the employed model receives an additional validation.

B. Conclusions

The integration of all the mentioned modules within a single Raspberry Pi was a complex task (as many incompatibilities occur when trying to interconnect the multiple devices) however, the integration was successful and the system showed its functionality. The final model provided us with an astounding tool that aims to provide a complete report on body biometrics, such as the ones in the brain via a EEG (OpenBCI) and body (Empatica, Hexiwear).

Using the ML model created, the SHAP Fig. 7 was created, which provides the significance of each feature regarding mental fatigue's classification. The plot reveals that high values of features from P7 and P8 channels are correlated with low levels of fatigue, in contrast to the C4 channel, in which high values of γ_{C4} are correlated with lower levels of fatigue.

The methodology followed within the project was concise and flexible, as it could be further escalated with additional biometrics, as well as algorithms that could enhance the website created. Although the system was not tested on enough participants, nor it was used in a field study within a real classroom (due to the current pandemic), we believe that the current development has the potential to serve as a basis to further revolutionize the current educational system that employs traditional teaching.



Fig. 9. A participant using our interactive environment and biometric devices



Fig. 10. An investigator performing backend-related tasks

REFERENCES

- [1] M. NarayanMohanty and R. Sikka, "overview on fingerprint-based identification system," *Materials Today: Proceedings*, 2021.
- [2] A. H. Mohsin, A. A. Zaidan, B. B. Zaidan, O. S. Albahri, S. A. Bin Ariffin, A. Alemran, O. Enaizan, A. H. Shareef, A. N. Jasim, N. S. Jalood, M. J. Baqer, A. H. Alamoodi, E. M. Almahdi, A. S. Albahri, M. A. Alsalem, K. I. Mohammed, H. A. Ameen, and S. Garfan, "Finger vein biometrics: Taxonomy analysis, open challenges, future directions, and recommended solution for decentralised network architectures," *IEEE Access*, vol. 8, pp. 9821–9845, 2020.
- [3] V. Garg, A. Singhal, and P. Tiwari, "a study on transformation in technological based biometrics attendance system: Human resource management practice," in *2018 8th International Conference on Cloud Computing, Data Science Engineering (Confluence)*, pp. 809–813, Jan 2018.
- [4] J. R. Anderson, S. Betts, J. L. Ferris, and J. M. Fincham, "Neural imaging to track mental states while using an intelligent tutoring system," *Proceedings of the National Academy of Sciences*, vol. 107, no. 15, pp. 7018–7023, 2010.
- [5] K. W. Brawner and A. J. Gonzalez, "Modelling a learner's affective state in real time to improve intelligent tutoring effectiveness," *Theoretical Issues in Ergonomics Science*, vol. 17, no. 2, pp. 183–210, 2016.
- [6] J. C. Y. Sun and K. P. C. Yeh, "The effects of attention monitoring with EEG biofeedback on university students' attention and self-efficacy: The case of anti-phishing instructional materials," *Computers and Education*, vol. 106, pp. 73–82, 2017.

- [7] P. W. Kim, "Ambient intelligence in a smart classroom for assessing students' engagement levels," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, pp. 3847–3852, Oct 2019.
- [8] R. Wass, J. Timmermans, T. Harland, and A. McLean, "Annoyance and frustration: Emotional responses to being assessed in higher education," *Active Learning in Higher Education*, vol. 21, no. 3, pp. 189–201, 2020.
- [9] J. Gruzelier, M. Foks, T. Steffert, M.-L. Chen, and T. Ros, "Beneficial outcome from eeg-neurofeedback on creative music performance, attention and well-being in school children," *Biological Psychology*, vol. 95, pp. 86–95, 2014. SAN (Society of Applied Neuroscience) Special issue on Neurofeedback.
- [10] J. Xu and B. Zhong, "Review on portable eeg technology in educational research," *Computers in Human Behavior*, vol. 81, pp. 340–49, 2018.
- [11] A. Fletcher and C. Mura, "Ten quick tips for using a Raspberry Pi," *PLoS Comput Biol*, vol. 15, p. e1006959, 05 2019.