CSI 4900 - DreamCraft AI

Trinity Vermeire
300129927
Faculty of Engineering
University of Ottawa
tbate032@uottawa.ca

Bradley Wills
300187635
Faculty of Engineering
University of Ottawa
bwill040@uottawa.ca

Abstract

This report describes the creation of an innovative project that uses a mobile application and an EEG (Electroencephalography) device to enhance the quality of a person's sleep. The objective of this project was to develop an extensive deep learning model that uses real-time EEG data to generate binaural audio meant to specifically encourage and induce slow-wave and REM (Rapid Eye Movement) sleep cycles. The real-time EEG monitors brainwave activity by using electrodes, an Arduino UNO, and a Bioamp EXG Pill. The frontend application was created using the Flutter framework for cross-platform compatibility, and Python was utilized for the backend processing. This project offers an innovative solution to ameliorate sleep patterns for improved health and wellness, and demonstrates how to combine multiple branches of software and hardware technologies to aid in sleep-related issues.

I. Introduction

Sleep affects mental clarity, emotional regulation, and general quality of life, all of which have a substantial impact on human health and wellness [1]. Although it is important to get enough sleep, many people experience sleep related issues like insomnia and irregular sleep cycles, which can be harmful to their mental and physical health. The approach taken on during this project is to use binaural audio to entrain the brain with low frequencies that promote the most health beneficial deeper sleep stages. Research indicates that entraining brainwaves and promoting relaxation can be achieved through the use of binaural audio, which are produced by two slightly offset frequencies playing in each ear at the same time [2]. As the amount of sleep needed varies from person to person, it is crucial that a solution is defined by the individual and actively learns as it is implemented [5]. With the use of a deep learning algorithm that implements the LSTM (Long Short-Term Memory) and CNN (Convolutional Neural Networks) models, adaptive learning and individual results are possible. By incorporating this process into a simple and sleek mobile application, this method of uniquely enhancing sleep quality will be readily available to all. This implementation may help people who have trouble falling asleep by lowering the frequency of their brain, inducing slow-wave and REM sleep at a faster rate, and produce an overall high-quality restful and restorative sleep.

II. Literature Review

A. Sleep Stage Analysis

Sleep stage analysis is the process of categorizing different stages that occur during sleep based on unique patterns of brain activity and physiological changes. There are five sleep stages that are entered during a sleep event: wake, N1, N2, N3, and REM.

People exhibit conscious awareness when they are in the wake state, they produce alpha waves when their eyes are closed and beta waves when they are open. This is a demonstration of the level of activity in the brain corresponding to which brainwaves are exhibited, with the lower frequencies being produced with less visual stimulation. The first sleep stage the body enters is Stage N1, which lasts approximately one to five minutes and is characterized by theta waves and the presence of muscle flexion. Next is Stage N2, this phase lasts for approximately twenty-five minutes and is marked by K-complexes, deeper sleep, sleep spindles, and a drop in body temperature and heart rate. Afterwards, one enters slow-wave sleep or Stage N3, it is the deepest non-REM sleep stage and displays delta waves. It functions as a high arousal threshold and is necessary for tissue growth and repair in the immune system. The phenomenon of dreaming is linked to REM sleep, which is distinguished by beta waves that resemble wakefulness and atonic muscles other than the diaphragm and eyes. Each REM cycle lengthens during the night and is accompanied by disorienting muscle movements and heightened brain activity [3].

As Stage N3 and REM sleep hold the greatest plethora of health benefits, it is the goal of this project to promote their occurrence during the process of sleep. It is essential to understand these stages of sleep and their characteristics for accurate sleep stage analysis that allows insights into sleep quality and the related physiological processes.

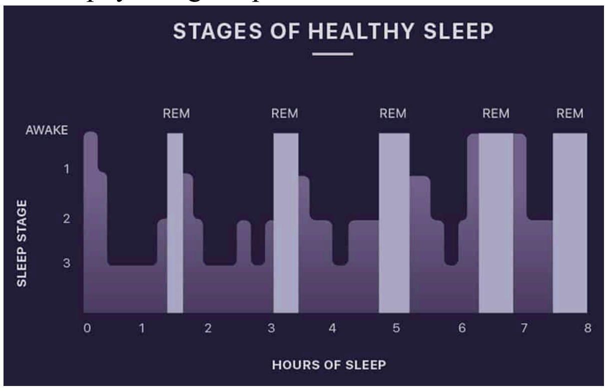


Figure 1: Sleep Stages Throughout the Night [19]

B. Effects of Binaural Audio Stimuli

A type of auditory stimulation known as binaural audio involves presenting two tones at different frequencies to each ear separately. This causes the brain to perceive a third frequency, which is referred to as the binaural audio. This third sound is the result of the brain taking the two separate audio stimuli and merging them to create one audio event. The act of merging these audio stimuli makes the brain an active participant in the creation of this stimulation, and therefore it has a greater impact on the brain than a non-binaural approach. This phenomenon is observed in the 1-30 Hz frequency range, which corresponds to the human EEG frequency bands [4].

It is believed that delta binaural audio influence brainwave activity by promoting a calm demeanour and optimizing relaxation. Their approximate frequency range is 1-4 Hz, which is also the frequency attained during Stage N3 of sleep, when brain activity is at its lowest. Clinical trials suggest that listening to delta binaural audio can induce a calming state that is beneficial for falling asleep by causing people's brainwaves to synchronize with the audio frequency [2]. This stimulation with delta binaural audio can enhance sleep in several ways, including reduced sleep latency, fewer awakenings during the night, longer sleep duration, and enhanced quality of sleep. Moreover, individuals who awaken from sleep induced by delta binaural audio often report feeling happier and more rested [2]. By aiding the brain with these frequency changes, it creates a seamless transition into the lower frequencies, which are very contrary to the high activity of wakefulness.

C. Brainwave Frequency Analysis

In a similar demeanour to sleep classification, our brainwaves can also be categorized into different bands, all with their own significance and use during our lives. Brainwaves are categorized as synchronized electrical activity within the brain, and are the result of sets of neurons firing at once. When many neurons send electrical pulses at the same time, it can be detected by an EEG device using sensitive electrodes located at specific spots on the head [6].

There are five brainwave categories as seen in Figure 2. The first category is that of Gamma frequency, which ranges from 32-100 Hz. This represents a state of high awareness and shows fast activity in the brain that can be related to problem-solving and learning. The next band is Beta frequency, which ranges from 13-32 Hz and constitutes normal levels of awareness and active thinking, it is associated with tasks like active focus and conversation. Alpha frequency occurs within the band of 8-13 Hz and is seen when the brain and body are in a state of relaxation, often found during activities like yoga and artistic creation. The next band contains Theta frequency ranging from 4-8 Hz. They are produced when a person is deeply relaxed and exhibits reduced levels of consciousness, as seen during tasks like meditation. The final band of brainwave is that of Delta frequency, ranging from 1-4 Hz, which is the lowest frequency exhibited in the brain. It is experienced during deep sleep while the body and mind are in a state of healing [6].

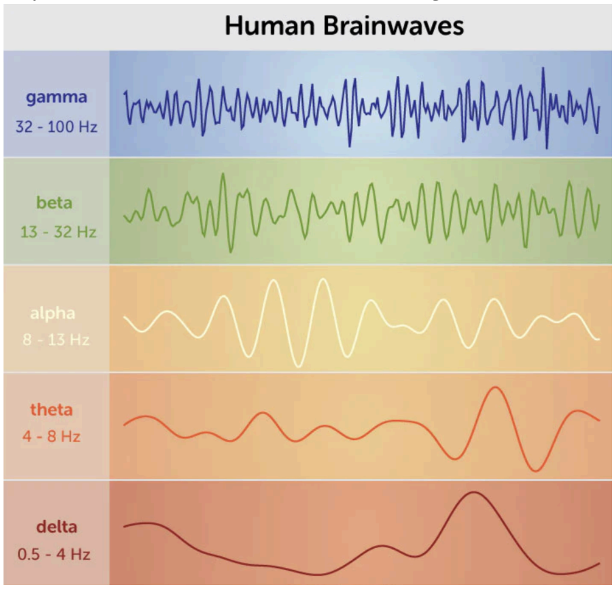


Figure 2: Human Brainwave Patterns [6]

D. *Deep Learning for Sleep Classification*

The process of classifying sleep stages from abstract EEG data requires an extensive and meticulously constructed deep learning model. Deep learning is classified as the use of multi-layered neural networks that are used to simulate the complex decision-making powers of the human brain [9].

To accurately classify such a data intensive task, the learning model will need to contain many hidden layers between the input and output that are used to learn intricate structures, allowing it to predict the current and future sleep stage of a participant. These hidden layers, as seen in Figure 3, contain a plethora of interconnected neurons which work by applying weights to the input data and passing them through an activation function. The application of this function allows the model to learn non-linear dependencies between input and output data and accurately identify underlying patterns [10].

Each layer is connected to the next, with each one passing on the weighted inputs. In turn, this forms a complex pattern, with weights being learned across the system as it is trained. The more hidden layers that are used, the more complex a pattern the model can learn given a sizable dataset and overfitting detection. These extensive layers allow the model to accurately perform feature extraction, which is the process of analyzing relevant information to make an educated prediction [10]. This allows a style of learning similar to the human experience, where the interconnected nodes in the neural network can be perceived as neurons in the brain.

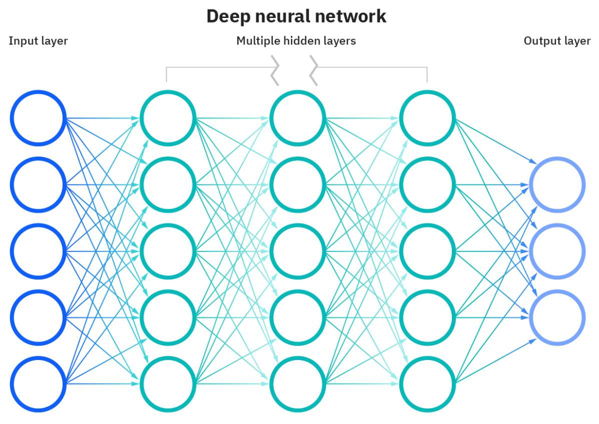


Figure 3: Deep Learning Model [11]

III. Methodology

A. *Training Dataset*

The Sleep-EDF dataset from PhysioNet containing sleep cycles from 197 participants was used to train the LSTM-CNN deep learning model [7]. This data was organized into 2 files per night of sleep, a HYP file and an PSG file. The HYP files contain two columns with data of the time span and the corresponding sleep stage that occurred within it. The sleep stages are classified into seven categories: Sleep stage 1, Sleep stage 2, Sleep stage 3, Sleep stage 4, Sleep stage R, Sleep stage W, and Movement time. The PSG files contain 5 columns, these include the event markers for a change in sleep stage as well as voltages received from four electrode channels corresponding to the following: EEG Fpz-Cz (Frontal Pole to the Central Zero on the midline), EEG Pz-Oz (Parietal midline to the Occipital midline), EMG Submental (muscle activity below the chin), and EOG Horizontal (eye movements). This data was recorded with a sampling rate of 100Hz and contains data before and after sleep occurs [7].

B. *Training Data Preprocessing*

In order for the deep learning model to accurately classify the data being provided by our EEG, the model must be trained on data of a similar distinction. The data given from the PhysioNet Sleep-EDF has the classification separated into two files: HYP and PSG [7]. To use this data within the deep learning model, the information had to be consolidated to one file that contains both the sleep stage and the frequency at that time.

Since the sleep stages used in the HYP files do not conform with our chosen classification of sleep stages, they must be processed. Sleep stage 3 and 4 are essentially the same sleep event with stage 4 simply being a continuation of stage 3, as such the two labels were merged to both be classified as Sleep stage 3. As our EEG is unable to detect active motion, the Movement time category was merged with Sleep stage W (wakefulness) as they both indicate an active and non-relaxed body [7].

The PSG files contain 4 channels sampled every 0.01 seconds, while our EEG only has one channel but can have the same sample rate for simplicity purposes [7]. Due to this, the training data was modified to only contain one channel (EEG Fpz-Cz) and have one entry/row for every 1 second. There was also a massive over-representation of Sleep stage w within the dataset, with over 50% of the entries being such. As this skewed data would severely hinder the training of a deep learning model, a proportional fraction of these entries were removed from each file. Through a Python algorithm, the data was processed to contain 2 columns: the EEG Fpz-Cz frequency with a sampling rate of one second and the corresponding sleep stage; this formatted data was exported as a set of cohesive CSV files.

C. Voltage Data Processing

What the electroencephalogram does is quite simple in the case of our hardware. It starts by capturing the voltage of your body through the reference electrode. This electrode is placed in a spot which is mostly void of nerves (such as the earlobe) and its purpose is to acquire your body's overall voltage. Then, the same process is also used for the scalp electrodes. The scalp voltages are then individually run through a differential amplifier alongside the reference voltage to get the difference between the two. These differences are then averaged, and thus you have your amplitude value [18]. These amplitude values are still in the form of microvolts, and much processing is still required to be able to analyze the data. In the case of detecting heartbeats or muscle flexions, you only need the microvolts and a spike is determined to be a flex. For our purpose, however, we need to capture the frequencies of the brain and thus, the sensitivity is much higher. The most typical way of converting from these microvolt values to a frequency is via a Fourier transform. This spreads the frequencies and their powers across a spectrum. This spectrum can then be analyzed to produce a singular frequency value.

D. Algorithm Development

During the initial development of the deep learning model, there was an attempt to use a fairly simple approach without the implementation of deep learning techniques. This was enacted by implementing LSTM layers along with dense layers and a low dropout rate. As there were minimal learning functions used and a lack of hierarchies, the model did not have the capabilities to learn the complex relationship between frequency and sleep stage. This was mitigated by updating the model to contain blocks of different neural network layers to optimize the learning capabilities and allow pattern recognition to occur.

The outputs to the LSTM model are exclusively used in the reward function for the audio output reinforcement learning model. We collect the most recent epoch (30 seconds) of data and send that to the LSTM, which returns the current sleep stage. Immediately after, we send the current frequency, sleep stage, and number of minutes until your alarm to the reinforcement learning model, which happens to have the architecture of a recurrent neural network. This should be enough information for the model to be able to accurately decide what audio stimuli is best for you. What this model deems as “best” is determined by a reward function, with the sleep stage and time until alarm as the main parameters. Throughout the majority of the night, the reward function is meant to prioritize maximizing time in deep sleep and REM. In order to implement the wake-aid, when your alarm is nearing, it linearly transitions to rewarding lighter or waking stages of sleep and punishing for deep stages of sleep.

E. Data Collection

The data collected from our EEG device is saved in the form of CSV files, with one row being attributed to one second of data. The columns are the following: UserID, delta, theta, alpha, beta, frequency, and timestamp. Through analysis, it was determined that these categories would be sufficient for any future data requirements. The delta, theta, alpha, and beta columns hold the power values associated to those frequency ranges, which prove to be highly useful when it comes to sleep stage prediction. The frequency is a heuristic brainwave prediction found by calculating the arithmetic weighted mean of numerous bands of frequencies between the ranges of 1-30 Hz and their associated powers. When we graph this frequency value throughout the night, we do see the general progression of the sleep stages as our brainwaves rise and dip accordingly. As seen in figures 5 and 6, these graphs have similarities to the training dataset.

F. Testing and Evaluation

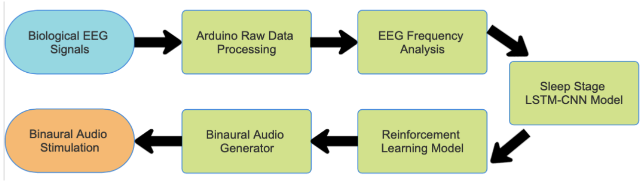
As we have yet to properly train the reinforcement learning model, we have not been able to perform any testing or evaluation of the fully completed application. We are in the process of running the software throughout the night, but the reinforcement learning model does not learn very efficiently (which is an inherent issue with biological signals, as the action taken from the model does not immediately affect the next state). We are at least a week away from producing any good results. Even with the results, we wouldn't be able to easily compare it to a normal night's sleep, as we didn’t collect data from enough normal nights to be able to generalize and produce average values.

IV. Implementation

A. Software Components

The flow of data for this system's architecture contains 5 distinct checkpoints, as shown in Figure 3. These stages are the following: input EEG signals, Arduino raw data processing, EEG frequency analysis, sleep stage LSTM-CNN model, reinforcement learning RNN model, binaural audio generator, and finally the binaural audio stimulation.

The input EEG signals are the raw data that is received using electrodes and the Bioamp EXG pill. This constitutes the voltage captured by three electrodes, two placed on the forehead and one underneath the ear. This data is then passed to the Arduino, where it is sent through an open serial port to be received by a computer. The voltage values are promptly converted to frequency values through the use of Fourier transforms, and are then passed to the LSTM-CNN sleep classification model. This model takes the last 30 seconds of EEG frequency data and outputs the predicted sleep stage, which was learned through the training dataset. The output sleep stage is transferred to the reinforcement learning model where based on the given state, an action is selected to generate an output representing the frequency for optimal sleep. After the action is enacted, a reward is created, serving as feedback on how well the action aligns with the objective of achieving optimal sleep. The feedback from the reinforcement learning model is then sent to the binaural audio generator, where it creates two audio clips of a frequency with a predetermined offset. Finally, these two audio clips are played in separate ears and binaural audio is able to entrain the brain for an optimized sleep.



Biological EEG Signals
Arduino Raw Data Processing
EEG Frequency Analysis
Sleep Stage LSTM-CNN Model
Reinforcement Learning Model
Binaural Audio Generator
Binaural Audio Stimulation

Figure 4. Flow Chart of the System Architecture

B. Hardware Components

Finding appropriate hardware was quite difficult. There are many EEG headsets on the market, but very few satisfy our requirements. We wanted a one-channel, affordable EEG which didn't have any API and could easily send the raw data to our laptops. This task was very difficult and took months to actually get them put together. At one point, we even planned on buying all the electrical components and fully assembling them. We eventually found the BioAmp EXG pill and immediately ordered them. It took about a month and a half to ship, however, because the company who manufactures them happened to be starting a stock upgrade as soon as we placed the order, and they are also situated in India.

C. Performance Considerations

There were a few concerns which arose during development and are still persistent. To start, the frequency analysis results in a single frequency value, which we were expecting would eliminate a lot of the noise (especially considering we got these frequencies from 5 seconds worth of amplitudes). Contrary to our expectations, when we graph out a night's worth of data, it is very noisy and the sleep stages aren't very visible. Below, are a few samples of data captured from a normal night:

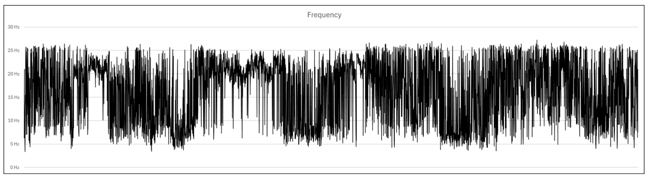


Figure 5. Line Graph of Brain Frequency

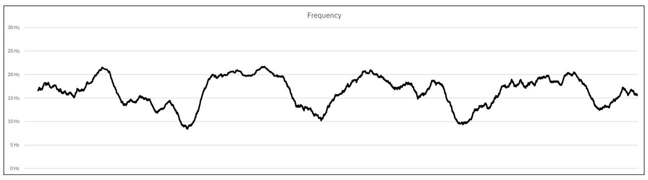


Figure 6. Trend line of Brain Frequency

We also have concerns with the accuracy of the sleep stage model. It was trained on a database of EEG data of regular, non-stimulated sleep. Thus, there is a potential that by listening to binaural frequencies, we may be messing with our brainwaves in a way that the model does not know how to interpret properly. This could lead to wrong sleep stage predictions and skewed metrics such as time spent in REM or deep sleep. Perhaps the reinforcement learning model will even learn a way to play odd frequencies which use our brain as an interface to manipulate the sleep stage model into rewarding it, while potentially hurting our sleep. If this problem arises, there is very little we can do. It would be incredibly difficult to re-train the sleep stage model, as there are no datasets which show how frequencies affect your brain-waves while asleep. The only workaround would be utilizing another sleep stage prediction method, such as radio waves [15] which doesn't rely on any measurements of brainwaves.

IV. System Architecture

A. Programming Languages and Tool

For the frontend, we decided to use Google's UI development software called FlutterFlow. This choice was primarily made because of the cross-platform capabilities of flutter as it can be run on Windows, Linux, macOS, web, and Apple/Samsung mobile devices [14]. We did not use any extra libraries or tools in our frontend, since our backend performed the majority of the processing.

As for the backend, we decided on Python as our primary language. This choice was made because of our extensive experience with the language and the numerous machine learning libraries. Some tools which we ended up using were the following:

- Numpy: math related to the Policy-Based Optimization Reinforcement Learning Model. (Secondary Option instead of a RNN)
- Tensorflow: all the neural networks.
- Pyaudio: playing the audio from binaural byte lists and mp3 files.
- Threading: multithreading to be able to create and play the audio in the background.
- Eeglib: splitting the FFT spectrum into multiple bands.
- Scipy.fft: performing a fast Fourier transform on the EEG data.
- Serial: interacting with the serial interface of the Arduino UNO and receiving data.
- CSV: saving the data in the morning after each sleep.
- Time: general time management and recording timestamps alongside the data in the CSV files.
- Keras: implementation of the deep learning model. Provided layers, evaluation metrics, and functions to aid in the progression of the training process.

B. *Deep Learning Model Architecture*

The deep learning model used for sleep stage classification is composed of the following five categories: input data, Block 1, flatten layer, Block 2, and Block 3 [7]. The input data fed to the model are blocks of EEG frequency data, with a duration of 30-seconds. This is sent directly to the Block 1 architecture where it passes through five different layers that are as follows: Conv2D, Batch Normalization, ReLU, Max Pooling, and Dropout.

The Conv2D layer uses a mathematical convolutional operation to apply weights to the input data, creating a feature map containing relevant patterns at different data-relevant spatial locations [12]. The feature map is then passed to the Batch Normalization layer, where the weights are normalized using the mean value and standard deviation in order to prevent vanishing gradients [13]. Vanishing gradients are defined as the backpropagation error signal being unable to reach the beginning of the neural network, this can severely impact the learning ability of a neural network as it is unable to learn from its errors [13]. The next layer of the deep learning model is ReLU, which has the same function as Batch Normalization with the minor difference of having constant and bigger gradients, which again minimizes the occurrence of vanishing gradients [13]. After the feature map has been normalized, it is passed through a Max Pooling layer where the maximum values in two unit regions are reduced to a single value, effectively reducing the size of the feature map [12]. Finally, the Dropout layer receives the feature map and prevents overfitting by randomly setting a percentage of neurons to the value 0, this keeps the network from curating itself solely for any specific feature [12].

The architecture of Block 1 is executed twice in series and then passed to a flattening layer where the input data is transformed into a one-dimensional vector. This vector is then sent to Block 2, which contains the following layers: Conv2D, Batch Normalization, and ReLU. This block is enacted to continue the back propagating learning process while ensuring the normalization of data.

The final sequence of layers the data is passed through is Block 3 which contain LSTM and softmax layers. The LSTM layer is unique, as it has the ability to 'remember' information that has been deemed as important while 'forgetting' irrelevant information [17]. This is possible with the implementation of the following four gates that exist within the layer: Forget Gate, Input Gate, Input Modulation Gate and Output Gate [17]. The Forget Gate is combined with the previous output to generate a number between 0 and 1, indicating how much of the previous state should be preserved, with 0 being none and 1 being preserving all [17]. The Input Gate also uses a value between 0 and 1 to determine what information to store in the LSTM cells. It does this by implementing a tan h block, which creates new values that must be engineered into the previous state [17]. The next gate in the LSTM model is the Input Modulation Gate, which is used to add non-linearity to the information by implementing a zero-mean requirement, allowing for faster convergence and reduced run time [17]. Finally, the Output Gate decides what information is to be produced as a result, it considers the previous and hidden cell states and chooses the most probable one to be propagated as output to the next layer [17]. The softmax layer is used to map the input signals to the output signals, this occurs since the number of units in this layer correspond to the number of distinct features [12]. To conclude this operation, a single output is produced containing the feature predicted from the input, indicating the assumed sleep stage.

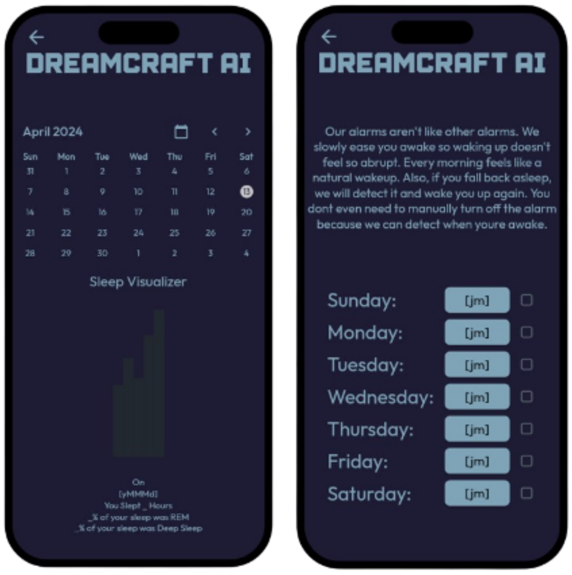
C. *Mobile App Development*

For the mobile application, we decided to use Google's UI development software called FlutterFlow. This choice was primarily made because of the cross-platform capabilities of flutter as it can be run on Windows, Linux, macOS, web, and Apple/Samsung mobile devices[14]. The development process was quite simple, but it took me a few weeks to get it all polished. There is a main home page with 6 pages that you can be redirected to. These buttons to redirect you are encapsulated in a dropdown box which when it is collapsed, displays the amount of time until your alarm will sound. The secondary pages are as follows:

Settings: This page includes a few settings options which the user might want access to. The notifications setting is quite self-explanatory. When it is toggled on, the app can send you notifications. These notifications would be reminders of when you should commence trying to sleep. In order to dynamically calculate when you should go to bed, it would take your alarm time, subtract your Sleep Goal value, and subtract the average amount of time it usually takes you to fall asleep. Thus, when you listen to the notification, you are likely to achieve your sleep goal before you have to wake up in the morning.

The SleepAid determines whether you want the app to try to play audio stimulation while you are awake. This would have the goal of bringing you to sleep faster than you naturally would.

The Sleep Goal is the amount of time you want to be asleep for. Some people have disorders where they might need more or less sleep than the average person, so we decided to have this be customizable. This value will only be used in the calculation for the notifications.


DREAMCRAFT AI

April 2024

Sun	Mon	Tue	Wed	Thu	Fri	Sat
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	1	2	3	4

Sleep Visualizer

On
[yMMMd]
You Slept _ Hours
_% of your sleep was REM
_% of your sleep was Deep Sleep

DREAMCRAFT AI

Our alarms aren't like other alarms. We slowly ease you awake so waking up doesn't feel so abrupt. Every morning feels like a natural wakeup. Also, if you fall back asleep, we will detect it and wake you up again. You dont even need to manually turn off the alarm because we can detect when youre awake.

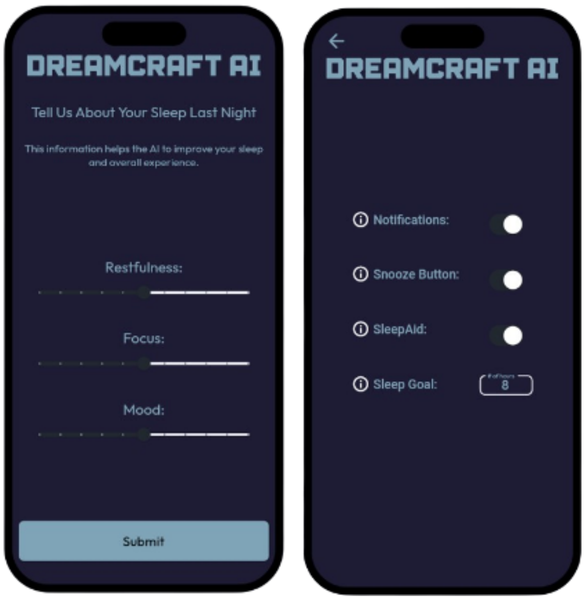
Sunday:	[jm]	☐
Monday:	[jm]	☐
Tuesday:	[jm]	☐
Wednesday:	[jm]	☐
Thursday:	[jm]	☐
Friday:	[jm]	☐
Saturday:	[jm]	☐

Figure 7: Sleep Visualizer and Alarm Mobile Application Pages

Sleep Visualizer: This page allows you to select any date in the past in which you used the app to assist your sleep. Once selected, this page will provide a visualization of the sleep stages you went through during that night, as well as a few statistics about that night's rest and even potentially whatever the user's feedback was for that day.

Alarms: On this page, the user is able to manipulate their alarms and activate/deactivate them depending on their preferences. There is only one alarm slot for each day of the week, as giving the user access to however many alarms may lead to problems. Nevertheless, this application is only meant for sleeping throughout the night. If the user wants to have a quick nap, we will most likely have a nap button on the home page which would allow the user to set a temporary timer/alarm in which they want to be woken up by. The app would still work the same and optimize their sleep no matter the length of the nap.

Feedback: This page permits the user to evaluate their sleep with a few metrics. There is a scale from 0 to 10 (inclusive) in which the user can rate their restfulness, focus, and mood. This data could be used to refine the reinforcement learning model or for the user to self-analyze the effects of their sleep on their day.


DREAMCRAFT AI

Tell Us About Your Sleep Last Night

This information helps the AI to improve your sleep and overall experience.

Restfulness:

Focus:

Mood:

Submit

DREAMCRAFT AI

Notifications: ●
Snooze Button: ●
SleepAid: ●
Sleep Goal: 8

Figure 8: Feedback and Settings Mobile Application Pages

Stimulation: Although this page was not completed, it will hold a lot of importance in the future of this product. On this page, we intend on having the subconscious messaging options as well as a few personalization options in regard to the types of stimulation the user may want. We are undecided on what these types of stimulation may be. Perhaps we could have nature/ocean sounds, actual AI ambient music generation, or simply binaural frequencies.

Profile: This page was also not completed, and it may never be completed, as we are undecided on the necessity of it. This page would serve a purpose only if we decide to lean into the potential of this app as a social platform. You would theoretically be able to share your previous sleeps, your subconscious messages, or even recordings of sleep talking.

D. *Integration of Backend and Frontend*

Very little integration was needed to be done as of the current stage of our project. The majority of the pages in the application (frontend) do not have any benefit of being connected to the backend. All we needed to do was to make it so that when the application gets opened, the main python file (serial manager) is run. Also, we had a few shared data files used between the backend and frontend, such as the sleep data (for visualization purposes) and the alarm times (for the wake-aid in the backend). We plan on using HTTP requests to communicate live data, such as notifying the backend of an update of the alarm times.

E. *Testing and Debugging*

There was quite a lot of testing and debugging we had to do for this project. Because of how new Flutter is, we ran into a few bugs while creating the frontend. Fortunately, most of them were not that complicated to solve. One that is still persistent is showing the amount of time until the alarm. Strangely, when you collapse the dropdown box, the time doesn't show up immediately, and you need to open and close the dropdown box a second time before it becomes visible. Also, the passage of time seems a bit odd, as it appears to be accelerated when you have the dropdown box opened.

The backend was the culprit for the majority of the testing/debugging. Considering neither of us have any prior experience in sleep science or electroencephalograms, we had to learn a lot. A significant portion of that learning was from trial and error.

To start, UpsideDownLabs recommended pairing their Bioamp EXG pill with another chip called an AdaFruit Raspberry Qt Pi to do the computing and conversion of the data to send to the laptop. It took a long time and a lot of testing to discover that this chip is simply incompatible with our laptops and that when we plugged them in, we couldn't detect any open serial ports. Thus, our solution was to instead use the well known Arduino UNO. This worked perfectly, and we could now access the EEG data.

The next issue we encountered was to do with frequency analysis. We initially structured our code to extract the brain's frequency via calculating the average distance between local peaks. This consistently produced undesirable results and led us to do more research. We soon discovered that the ideal way of extracting frequency was with a Fourier transform. This works splendidly and results in accurate frequency values which seemingly follow a sleep cycle throughout the night. Unfortunately, this data is very spiky (Figure 5) and would most likely be difficult for the models to interpret so in the near future, I suspect we will end up implementing a polynomial regression algorithm (Figure 6) to simplify the data to a more straightforward trend line.

The remaining issues were all in the audio player/generator. At first, we attempted to have the transitions between frequencies be linear, but very quickly we discovered that changing the trend of the frequency even slightly created very abrupt transition noises that would most likely disturb the user's sleep. We solved this by using a half-period of a sine wave to move between frequencies and ensure the beginning and end of each transition had a slope of 0. The result was quite perfect and produced very soothing transitions. The few issues that persist are generally related to the audio quality not being up to par. There is occasional clipping between audio files and in the player, the decrease and increase of volume causes the speakers to produce unpleasant noises which somewhat resemble static. These issues won't be easy to fix, as they are inherent issues with the libraries we used for the audio.

V. Results

A. *Performance Evaluation*

Although we do not have sufficient data from both normal sleep and stimulated sleep, we can speak about how we would evaluate the performance if we were to have around 10 nights worth of sleep for both circumstances. There aren't any individual metrics to summarize the quality of a sleep, so we simply have to choose a few, and not all of them are quantitative. The metrics we would use are the percentage of time spent in rem and deep sleep, how long the sleep was overall, how many times the user was woken up during their sleep, how easy the wake-up was, and lastly, how refreshed the user felt. These would be compared via averaging out the values of the normal and stimulated sleep and finding the differences between the two. These metrics should be enough to prove the benefits of this product.

B. *Challenges and Limitations*

During the creation of this project, there were many developmental curves that were experienced. At the start of the project we were quite optimistic that there would be a fully functioning product by the end of this semester, however once we began nearing the point of full application deployment we discovered this was not possible with the scope of our work. There are many moving parts that must all be fully functional to have a cohesive working product, which was not possible during such a small time frame.

One of the biggest learning experiences that occurred during this process is that of participating in extensive research before entering a complex task. Throughout the creative process, there were times when we underestimated the amount of background knowledge that was required to complete the task, however we started the work anyway. This caused us to have to back-track later on during development when we realized the extensive knowledge base that was required was not properly implemented, creating a higher workload. This was seen during the deep learning model development as well as the hardware development. The deep learning model initially was a very simple model that happened to have a relatively high accuracy, it wasn't until the data was properly analyzed that it was found to be incredibly unbalanced, causing the accuracy to be raised with little learning occurring.

Initially, we chose to use a premade EEG headset as hardware, however with further research we found that it had to be operated through the use of another companies API which was not applicable to our project standards. This caused a delay in receiving the new chosen hardware and ultimately hindered our timing for creating visual results of the product.

Another challenge was that of understanding the science behind our data, initially we understood the values from the EEG to be the frequency values, however we found them to actually be the voltages. With this new understanding, we had to modify our training data as well as our EEG data processing to create the frequency based information required.

VI. Conclusion and Future Work

In the future, we have plans on expanding this project into a real product which would instead use radio waves to interpret the user's sleep stage [15]. The actual implementation of this technology could be done two ways, either a box you mount on your wall or a ceiling fan above your bed with a thermal camera facing downwards. Both of these products would use radio waves to detect your sleep stage, but the ceiling fan would have the ability to manipulate your temperature. As for audio stimulation, they would have either speaker or Bluetooth options depending on the users' preference. These physical devices would be paired with an account or phone with the DreamCraft AI application. The rest of the framework would widely be the same, but we would have to fully integrate the backend with the frontend as well as completing the remaining pages of the mobile application. We intend on delving more into the sciences of both subconscious messaging and diagnosing or medicating specific disorders. Such disorders are insomnia, depression, or even Alzheimer’s. We can treat Alzheimer’s with the power of neurostimulation [16]. By playing gamma frequencies, we can stimulate the hippocampus and other regions associated with memory, thus strengthening the neuronal connections and improving memory. We also have plans on bringing the power of subconscious messaging to the public, which has not widely been done before. With knowledge of your sleep stage, these subconscious messages could be delivered at optimal times and could lead to decreases of bad habits such as drug use or gambling.

Overall, we think this product has a long way to go, and we intend on taking it there. We see a monetizable idea, and we think it has potential to be revolutionary. Nobody has combined these technologies in this way, and we are excited to be the first.

References

[1] H. Wein, “The benefits of slumber,” National Institutes of Health, https://newsinhealth.nih.gov/2013/04/benefits-slumber (accessed Apr. 22, 2024).

[2] R. Dabiri, M. R. Monazzam Esmaielpour, M. Salmani Nodoushan, F. Khaneshenas, and S. A. Zakerian, “The effect of auditory stimulation using Delta binaural beat for a better sleep and post-sleep mood: A pilot study,” Digital health, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9125055/ (accessed Apr. 22, 2024).

[3] A. K. Patel, “Physiology, sleep stages,” StatPearls [Internet]., https://www.ncbi.nlm.nih.gov/books/NBK526132/ (accessed Apr. 22, 2024).

[4] I. R. E. A;, “Binaural beats to entrain the brain? A systematic review of the effects of binaural beat stimulation on brain oscillatory activity, and the implications for psychological research and Intervention,” PloS one, https://pubmed.ncbi.nlm.nih.gov/37205669/ (accessed Apr. 22, 2024).

[5] C. C. medical Professional, “Controlled zzzs,” Cleveland Clinic, https://my.clevelandclinic.org/health/body/12148-sleep-basics (accessed Apr. 22, 2024).

[6] “A Deep Dive Into Brainwaves: Brainwave Frequencies Explained | Muse^{TM} EEG-Powered

Meditation & Sleep Headband," https://choosemuse.com/blogs/news/a-deep-dive-into-brainwaves-brainwave-frequencies-explained-2 (accessed Apr. 22, 2024).

[7] "Sleep-EDF Database Expanded v1.0.0," https://www.physionet.org/content/sleep-edfx/1.0.0/sleep-cassette/#files-panel (accessed Apr. 22, 2024).

[8] N. Bobra, "nerajbobra/lstm-cnn-eeg-sleep-staging," https://github.com/nerajbobra/lstm-cnn-eeg-sleep-staging (accessed Apr. 22, 2024).

[9] IBM, "What is Deep Learning?," *www.ibm.com*, 2023. https://www.ibm.com/topics/deep-learning (accessed Apr. 22, 2024).

[10] "Hidden Layer," *DeepAI*, May 17, 2019. https://deepai.org/machine-learning-glossary-and-terms/hidden-layer-machine-learning (accessed Apr. 22, 2024).

[11] S. Mohapatra, "Analyzing and Comparing Deep Learning Models," *Analytics Vidhya*, Nov. 18, 2022. https://www.analyticsvidhya.com/blog/2022/11/analyzing-and-comparing-deep-learning-models/ (accessed Apr. 22, 2024).

[12] O. Yildirim, U. Baloglu, and U. Acharya, "A Deep Learning Model for Automated Sleep Stages Classification Using PSG Signals," *International Journal of Environmental Research and Public Health*, vol. 16, no. 4, p. 599, Feb. 2019, doi: https://doi.org/10.3390/ijerph16040599. (accessed Apr. 22, 2024).

[13] L. Gupta, "Batch Normalization and ReLU for solving Vanishing Gradients," *Medium*, Apr. 26, 2021. https://medium.com/analytics-vidhya/how-batch-normalization-and-relu-solve-vanishing-gradients-3f1a8ace1c88 (accessed Apr. 22, 2024).

[14]"Multi-Platform," *flutter.dev*. https://flutter.dev/multi-platform (accessed Apr. 23, 2024).

[15]"Learning Sleep Stages from Radio Signals: A Conditional Adversarial Architecture," *sleep.csail.mit.edu*. http://sleep.csail.mit.edu/# (accessed Apr. 23, 2024).

[16]"Evidence that gamma rhythm stimulation can treat neurological disorders is emerging," *MIT News | Massachusetts Institute of Technology*, Jan. 18, 2024. https://news.mit.edu/2024/evidence-gamma-rhythm-stimulation-can-treat-neurological-disorders-emerging-0118 (accessed Apr. 23, 2024).

[17] "Long Short Term Memory Networks Explanation," *GeeksforGeeks*, Jul. 09, 2019. https://www.geeksforgeeks.org/long-short-term-memory-networks-explanation/ (accessed Apr. 23, 2024).

[18] Smith, E. J., *An introduction to EEG*. EBME. https://www.ebme.co.uk/articles/clinical-engineering/introduction-to-eeg (accessed Apr. 23, 2024).

[19] M. Saklaine, "Stages of Sleep Cycle: Simply Explained," *EduGonist*, Dec. 28, 2021. https://www.edugonist.com/stages-of-sleep-cycle-simply-explained/ (accessed Apr. 23, 2024).