Targeted lucidity reactivation implemented in an open source watchOS app

Rishabh Mallela[1], and Remington Mallett[2]

[1]University of Pennsylvania, Philadelphia, PA, USA
[2]Northwestern University, Evanston, IL, USA

Summary. The induction of lucid dreams in ecological settings is critical for a comprehensive understanding of their phenomenology, neural underpinnings, and feasibility for therapies. Recent methods have been developed to deliberately induce lucid dreams, but they are highly dependent on laboratory equipment. Namely, a method known as targeted lucidity reactivation involves pairing sensory cues with a state of mental reflection, tracking sleep stages using polysomnography, and playing sensory cues in REM sleep to induce lucidity. Playing cues during specific sleep stages is a critical component of targeted lucidity reactivation, and to-date there are very limited ways to derive sleep stages without polysomnography or proprietary wearables. To resolve this limitation and promote the testing of targeted lucidity reactivation in a variety of settings, we developed an open-source iOS/watchOS application that performs the entire targeted lucidity reactivation procedure (pre-sleep training, real-time sleep staging, and REM cueing). Critically, the app includes a custom real-time sleep staging algorithm to identify REM sleep using measures derived from the Apple Watch and accessible to any developer. The current study offers a technical framework for future research investigating the feasibility of inducing lucid dreams outside the lab using everyday technology.

Keywords: Lucid dreams, induction, application

1. Introduction

The exploration of lucid dreams, a phenomenon where individuals become aware that they are dreaming while still immersed in the dream world (Baird et al., 2019; LaBerge et al., 1981), has offered unique opportunities for therapy, scientific discovery, personal growth, and creative exploration. Recent laboratory methods have been developed to deliberately induce lucid dreams and aid in the promotion of such applications (Tan & Fan, 2023). One of the most convenient and reliable methods is targeted lucidity reactivation (TLR), where participants undergo a brief training period during which they learn to pair an audio stimulus with the notion of becoming lucid (Carr et al., 2023). Then, when participants are in rapid eye movement (REM) sleep, researchers play the same audio cue to induce lucidity. Though a leading approach to lucidity induction, more research is needed to determine the reliability of TLR, particularly how it compares to other induction methods (Tan & Fan, 2023).

The classification of different sleep stages is a fundamental aspect of TLR, as it is important to identify the occurrence of REM sleep. Sleep staging is traditionally performed by monitoring sleep in the laboratory with polysomnography, a suite of neurophysiological measures including brain, muscle, and respiratory signals (Markun & Sampat, 2020). However, the advance of wearable technology has introduced the capability of achieving sleep staging with minimal technologies in home settings. Wearable devices (e.g., FitBit, Oura) record physiological measures (e.g., heart rate, motion) to determine sleep stage with accuracy sufficient to perform sleep staging for most contexts (de Zambotti et al., 2018). These devices offer a valuable tool for researchers to track sleep in naturalistic environments (Mallett et al., 2023), but their proprietary nature places major limitations on certain use-cases. For example, companies typically offer a summary of the prior night's sleep upon awakening instead of real-time access to sleep staging. The inability to perform real-time sleep staging with wearable devices is a key limitation to inducing lucid dreams at home in the absence of polysomnography.

One approach to overcoming this limitation is to develop custom sleep staging algorithms based on raw measures derived from wearable devices. Some devices that do not provide direct access to sleep stages will still offer access to the relevant underlying measures, which can then be used by developers to stage sleep using custom algorithms. For example, Whitmore et al. (2022) developed an Android OS app to detect N3 sleep using motion and heart rate derived from a FitBit and utilized this setup to play sound cues and improve memory. Prior work has used similar measures derived from the Apple Watch for off-line sleep staging (Walch et al., 2019), but the algorithm cannot directly be applied in this context as it was designed for off-line, post data collection processing.

To offer a freely accessible tool to track sleep stages and induce lucid dreams on iOS devices, we developed an open-source iOS application that performs the complete TLR procedure including real-time sleep staging. We modi-

Corresponding address:
Remington Mallett, Center for Advanced Research in Sleep Medicine, Hôpital du Sacré-Cœur de Montréal, 5400 West Gouin boulevard (wing J, 5 e floor), Montreal, Quebec H4J 1C5, Canada.

Email: mallett.remy@gmail.com

Submitted for publication: November 2023
Accepted for publication: September 2024
DOI: 10.11588/ijodr.2024.2.100797

fied a previously validated off-line sleep staging algorithm that leverages motion, heart rate, and temporal information collected from an Apple Watch for real-time REM detection (Walch et al., 2019). The application offers the pre-sleep sound pairing step, and then plays cues upon detection of REM sleep, offering a full implementation of TLR for iOS users. All code to run the app is freely available in a public repository (https://github.com/rmallela26/TLR), including instructions for the server-side computing necessary for field data collection.

2. REM-detection classifier

Before development of the application, we first developed a REM-detection classifier for real-time sleep staging with measures derived from the Apple Watch. For development, we trained and tested the classifier using a publicly available dataset that includes simultaneous measures from the Apple Watch and polysomnography, as well as manually scored hypnograms that include sleep stages for the corresponding polysomnography data (Walch et al., 2019). We used a random forest classifier, which is an ensemble learning method that combines multiple decision trees to make predictions. Each decision tree in the random forest operates independently, and the final prediction is determined through a voting or averaging mechanism. The random forest classifier was implemented using scikit-learn (Pedregosa et al., 2011), set with default parameters unless otherwise noted *(min_samples_leaf* = 48).

Features were generated from raw data collected by the Apple Watch and input into the model every 30 seconds. The raw data consisted of heart rate sampled at 0.2 Hz, triaxial motion sampled at 30 Hz, and time in seconds. Heart rate data was included because it is known that average heart rate changes during periods of REM sleep (Cajochen et al., 1994). Motion data was included because participants who are in REM sleep experience sleep paralysis, or inability to move their limbs except for slight twitches (Peever & Fuller, 2017). Time data was included as the likelihood and length of REM periods increases throughout the night (Le Bon, 2020).

Heart rate was collected from Apple Watch sensors. First, the average of all recorded heart rates over the current 30 second epoch was determined. Then, we smoothed the average heart rate signal with an exponential moving average (EMA). The EMA value was then raised to the power of 3 and scaled down by a factor of 1000 to normalize the data. Tridimensional acceleration from Apple Watch sensors was given in units of g (1g = 9.8 m/s^2). All motion data up till the current point in time was subsequently interpolated to a frequency of 30 Hz and then converted to activity counts, a measure of acceleration within a time interval (Neishabouri et al., 2022). The magnitude of activity counts was calculated using vector addition, and the magnitudes pertaining to the current epoch were summed. This sum was squared and smoothed with an EMA. Lastly, the motion feature was obtained by normalizing the EMA. The time feature was converted from seconds elapsed since the beginning of the night to hours.

During model training, each epoch was labeled as 1 of 5 sleep stages: Wake, N1, N2, N3, or REM. Since the primary purpose of the algorithm was to detect periods of REM sleep, during testing, each epoch label was converted to either REM or “not REM” (Wake, N1, N2, or N3). To address class imbalance and improve REM detection, we applied a custom probability threshold of 0.24 for REM epochs when testing the classifier and cueing with the app. That is, the epoch would be classified as REM if the model predicted that the current epoch was REM with a probability of 0.24 or more (otherwise, not REM). This threshold was selected based on optimizing REM performance, especially recall, across various threshold options.

Classifier performance was evaluated using a leave-one-participant-out cross-validation scheme with measures of accuracy, recall, and precision. Accuracy refers to the proportion of all epochs correctly classified. Precision refers to the proportion of true REM epochs out of all predicted REM epochs, indicating how well the classifier falsely estimated an epoch as REM. Recall refers to the proportion of true REM epochs that were correctly predicted, indicating the classifier’s ability to identify all REM epochs. Although we

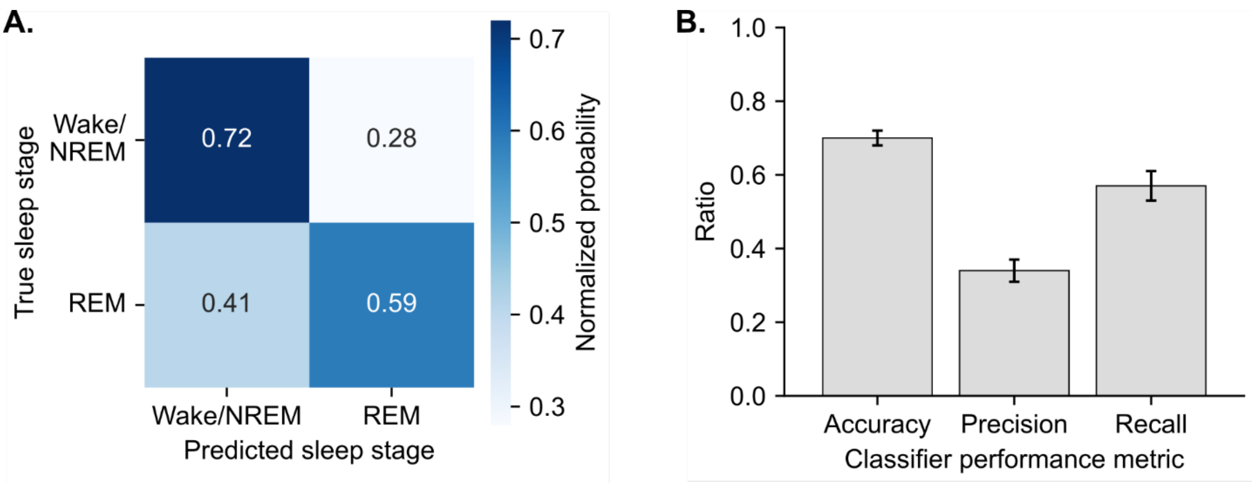

Figure 1. REM-detection classifier results. A) A confusion matrix indicating REM-detection performance and error tendencies. Cell values are normalized across rows. B) REM-detection performance evaluated with accuracy, recall, and precision. Scores are averaged across cross-validation folds. Error bars represent SEM.

designed the classifier to have the highest score possible across all three of these measures, recall was optimized at the expense of overall accuracy and precision. This was done because previous sleep studies suggest that it might be important to play cues at the onset of REM sleep, while the dream is still developing, as opposed to in the middle of the dream, when the dreamer is more fully immersed in the dream (Carr et al., 2023). The low precision was deemed tolerable as most errors were made in classifying N2 or N3 as REM, which are the deeper stages of sleep where the user most likely will not be disturbed by the cue.

Final validation results of the REM-detection algorithm are presented in Figure 1. The confusion matrix provides a descriptive account of general classifier performance, including which types of errors were most likely (Figure 1A). The mean classifier accuracy was 70%, mean classifier precision was 34%, and mean recall was 57% (Figure 1B).

3. iOS/watchOS Application

A shared application for watchOS and iOS was made to perform the entire TLR procedure (Figure 2). The application is designed to collect and transmit data to servers for processing, identify REM epochs, present TLR pre-sleep training, and present TLR sleep cueing (Figure 2A).

The iOS app (Figure 2B) receives the data and transmits it to the server every 30 seconds for processing using a post request. The server then responds, telling the application if the user is currently in REM sleep. Each time an epoch is classified as REM, the app plays the sound stimulus. However, if five or more consecutive epochs are classified as REM, the stimulus doesn't play until the REM period ends to avoid waking the participant. At the end of the night, a summary containing the number of times the sound stimulus played was presented.

When users press the "Training Audio" button on the iOS app, an audio recording is played that allows the user to become accustomed to the audio stimulus being used, and trains them to pair this audio cue with the notion of becoming lucid. The training audio, modeled after the TLR script provided by Carr et al. (2023), consists of a computer-generated voice that recites instructions for becoming lucid. During this pre-sleep training phase, a series of beeping tones (morse code for the word "dream") is included and the

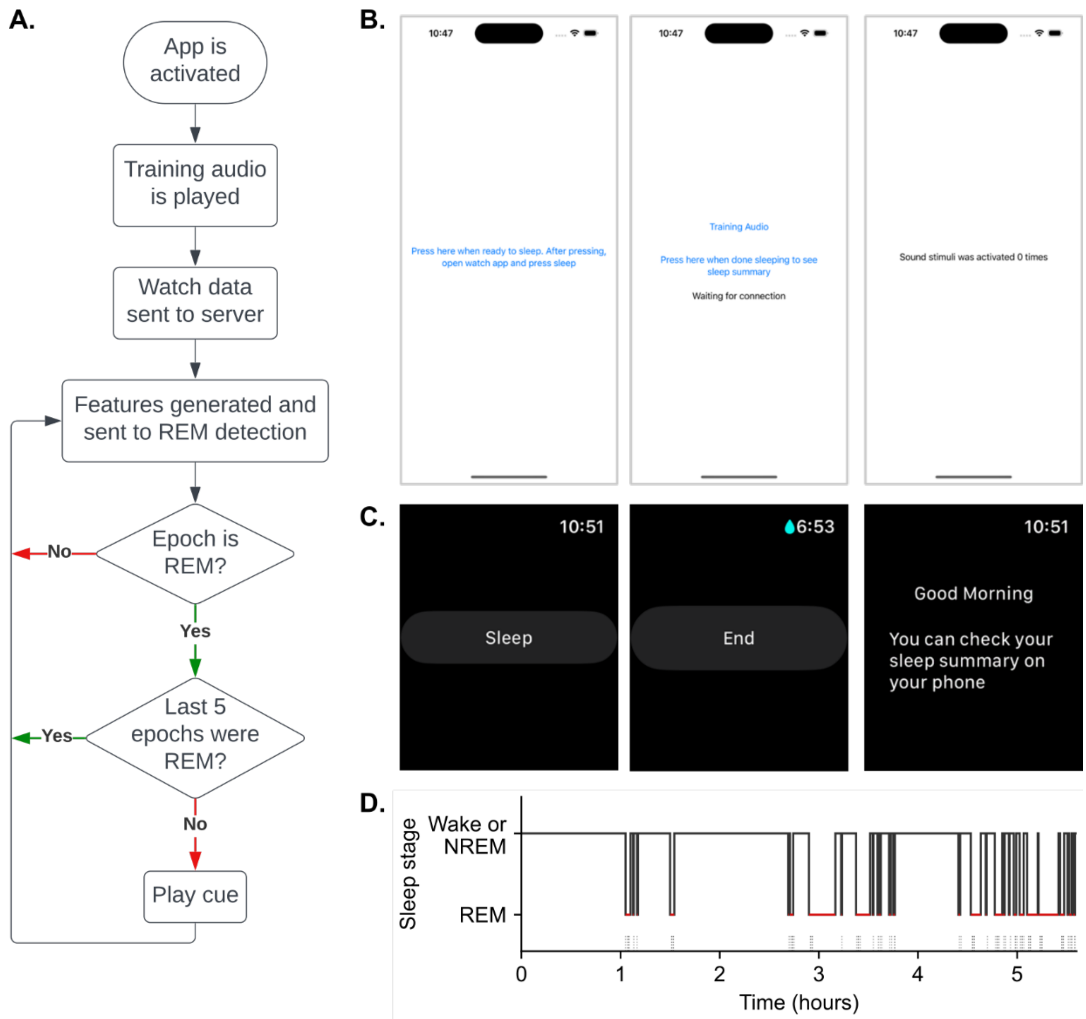

Figure 2. TLR app. A) Flowchart showing the application/server workflow. Data is transmitted to the server every 30 seconds. B) Screenshots of the iOS application. C) Screenshots of the watchOS application. D) A hypnogram and cue timestamps (dashed vertical lines) from a single session of using the application.

participant is told to associate this sound with becoming lucid. The application asks the participant to start training with their phone on maximum volume and then adjust the volume down as needed to find the loudest volume that will not wake them up. Towards the end of this audio, the user is expected to begin to fall asleep. Later, when the app detects that the user is in REM sleep, it will play the audio cue, thus sending a message to the user that they are dreaming. The watchOS app (Figure 2C) collects heart rate data through a workout session. It sends all data from the current epoch to the phone through a WatchConnectivity session for transmitting to the server to preserve the watch's battery life. The watchOS app also activates water lock to prevent the screen from being inadvertently tapped during sleep.

Data is exported to a server for processing. Whenever the app is used, the server saves the start time, the times when cues were delivered, the predicted label for each epoch, and all heart rate and motion data. All raw motion data and heart rates are saved on the server at the end of the night. Additionally, each epoch's score and the times during which the cues were played are also stored. Server data can be downloaded and loaded into the YASA Python package (Vallat & Walker, 2021) to generate a hypnogram and cue presentations (Figure 2D).

Usage instructions are available through the app and reproduced here:

1. Make sure your iPhone is charging and the Apple Watch is fully charged.
2. Unmute your iPhone and set it to max volume (you may adjust volume after the first night if it wakes you up).
3. Open the Dream Sage app on your iPhone and accept any requests asking to read data.
4. Click on the button saying "Press here when ready to sleep. After pressing, open watch app and press sleep". Do not exit the app for any reason or turn off the phone. If you do exit the app, you will need to restart these steps.
5. Open the Dream Sage app on Apple Watch and accept any requests asking to read data.
6. Click on the button saying "Sleep".
7. After 5 seconds, a water lock will enable automatically on the watch. This is to keep the screen from mistakenly being pressed while you are sleeping. If you want to exit the app for any reason, all you need to do is rotate the digital crown. If you exit the app after pressing "Sleep", you will need to restart these steps.
8. Click on the "Training Audio" button. This will play audio that will increase your chances at becoming lucid.
9. Put your iPhone on a nightstand next to you and lie down. Follow the instructions given in the audio.

4. Discussion

The development of this mobile application holds immense potential for researchers in the field. Traditionally, sleep studies involving large sample sizes have been restricted to lab settings, limiting the generalizability and ecological validity of the findings (Byun et al., 2019). With an open-source real-time sleep staging algorithm and lucidity-induction protocol, researchers might be able to conduct large-scale studies involving individuals in their natural sleep environments, leading to more representative and robust findings. This could enable researchers to gather comprehensive data on sleep patterns, REM sleep occurrences, and potentially explore the relationship between lucid-dream induction techniques and sleep physiology.

The main limitation of this application is its significant battery consumption on the Apple Watch. Since the Apple Watch must collect heart rates throughout the night, it must run a "workout session," which drains battery very quickly. While newer watches can handle this and last through the night, older watches with low battery health will not last the entire night. This will decrease the probability of lucid dream induction, as dreams occur more frequently towards the end of the night. This will also decrease the usability of the app, as people don't want to use apps that drain their battery and tend to not start the night with a full battery.

Another limitation of this application is the performance of the sleep staging algorithm, which could be improved. Because TLR relies on the ability to play cues during REM sleep reliably to induce lucid dreams, a highly accurate sleep staging algorithm is integral to the functionality of the app. Additionally, if playing cues during NREM periods wakes the user or reduces the quality of sleep, it would be counterproductive to the app. Although wearables offer accuracies that are generally acceptable, they are still less accurate than polysomnography and the value of their accuracies is largely dependent on the context of their use (de Gans et al., 2024). One of the main causes of this limitation in the current study was the lack of training data. The dataset used for training and testing for this algorithm consisted of only 31 subjects and one night of data for each subject. Additional limitations regarding the dataset are outlined in Walch et al. (2019).

A third limitation is data security. All the data collected by this app is stored on a server that can be accessed by the researcher leading the study. This introduces ethical and security related issues as this data is easily accessible at any time. Thus, study participants should be informed that their data is being collected, and when the study is done, a data security service should be put in place to ensure the privacy of user data.

The next step is to carry out a study to validate this application. Since TLR has already been validated and has worked in multiple contexts (Carr et al., 2023; Konkoly et al., 2021), we did not validate the application here. Other future directions could include improving specific parts of the app, such as the sleep staging algorithm and the user experience with the app. Additionally, the sensory stimulus used in this app can be changed if a more effective one is identified. Lastly, as the prospect of inducing lucid dreams outside the lab edges towards becoming a reality, research into therapeutic applications of lucid dreaming becomes all the more important. If it can be proven that having lucid dreams can help people, then apps such as this one can open up a whole new door of therapeutic options for patients (Holzinger et al., 2020; Ouchene et al., 2023).

5. Conclusion

In summary, this research demonstrates the feasibility of implementing real-time sleep staging and targeted lucidity reactivation outside the lab setting through the development of a mobile application. The main limitation of this application is the heavy usage of battery life, which limits the watches compatible with the app. The next step is to carry out a study to validate this application.

References

Baird, B., Mota-Rolim, S. A., & Dresler, M. (2019). The cognitive neuroscience of lucid dreaming. Neuroscience and Biobehavioral Reviews, 100, 305–323. https://doi.org/10.1016/j.neubiorev.2019.03.008

Byun, J.-H., Kim, K. T., Moon, H., Motamedi, G. K., & Cho, Y. W. (2019). The first night effect during polysomnography, and patients' estimates of sleep quality. Psychiatry Research, 274, 27–29. https://doi.org/10.1016/j.psychres.2019.02.011

Cajochen, C., Pischke, J., Aeschbach, D., & Borbély, A. A. (1994). Heart rate dynamics during human sleep. Physiology & Behavior, 55(4), 769–774. https://doi.org/10.1016/0031-9384(94)90058-2

Carr, M., Konkoly, K., Mallett, R., Edwards, C., Appel, K., & Blagrove, M. (2023). Combining presleep cognitive training and REM-sleep stimulation in a laboratory morning nap for lucid dream induction. Psychology of Consciousness: Theory, Research, and Practice, 10(4), 413–430. https://doi.org/10.1037/cns0000227

de Gans, C. J., Burger, P., van den Ende, E. S., Hermanides, J., Nanayakkara, P. W. B., Gemke, R. J. B. J., Rutters, F., & Stenvers, D. J. (2024). Sleep assessment using EEG-based wearables – A systematic review. Sleep Medicine Reviews, 76, 101951. https://doi.org/10.1016/j.smrv.2024.101951

de Zambotti, M., Goldstone, A., Claudatos, S., Colrain, I. M., & Baker, F. C. (2018). A validation study of Fitbit Charge 2TM compared with polysomnography in adults. Chronobiology International, 35(4), 465–476. https://doi.org/10.1080/07420528.2017.1413578

Holzinger, B., Saletu, B., & Klösch, G. (2020). Cognitions in Sleep: Lucid Dreaming as an Intervention for Nightmares in Patients With Posttraumatic Stress Disorder. Frontiers in Psychology, 11. https://www.frontiersin.org/articles/10.3389/fpsyg.2020.01826

Konkoly, K. R., Appel, K., Chabani, E., Mangiaruga, A., Gott, J., Mallett, R., Caughran, B., Witkowski, S., Whitmore, N. W., Mazurek, C. Y., Berent, J. B., Weber, F. D., Türker, B., Leu-Semenescu, S., Maranci, J.-B., Pipa, G., Arnulf, I., Oudiette, D., Dresler, M., & Paller, K. A. (2021). Real-time dialogue between experimenters and dreamers during REM sleep. Current Biology, 31(7), 1417-1427.e6. https://doi.org/10.1016/j.cub.2021.01.026

LaBerge, S., Nagel, L. E., Dement, W. C., & Zarcone, V. P. (1981). Lucid Dreaming Verified by Volitional Communication during REM Sleep. Perceptual and Motor Skills, 52(3), 727–732. https://doi.org/10.2466/pms.1981.52.3.727

Le Bon, O. (2020). Relationships between REM and NREM in the NREM-REM sleep cycle: A review on competing concepts. Sleep Medicine, 70, 6–16. https://doi.org/10.1016/j.sleep.2020.02.004

Mallett, R., Jameson, J. T., Paller, K. A., Markwald, R. R., & Russell, D. W. (2023). Clinical nightmare frequency and its association with reduced physical health during military operations. Sleep, 46(11), zsad214. https://doi.org/10.1093/sleep/zsad214

Markun, L. C., & Sampat, A. (2020). Clinician-Focused Overview and Developments in Polysomnography. Current Sleep Medicine Reports, 6(4), 309–321. https://doi.org/10.1007/s40675-020-00197-5

Neishabouri, A., Nguyen, J., Samuelsson, J., Guthrie, T., Biggs, M., Wyatt, J., Cross, D., Karas, M., Migueles, J. H., Khan, S., & Guo, C. C. (2022). Quantification of acceleration as activity counts in ActiGraph wearable. Scientific Reports, 12(1), Article 1. https://doi.org/10.1038/s41598-022-16003-x

Ouchene, R., El Habchi, N., Demina, A., Petit, B., & Trojak, B. (2023). The effectiveness of lucid dreaming therapy in patients with nightmares: A systematic review. L'Encéphale, 49(5), 525–531. https://doi.org/10.1016/j.encep.2023.01.008

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12(Oct), 2825–2830.

Peever, J., & Fuller, P. M. (2017). The Biology of REM Sleep. Current Biology, 27(22), R1237–R1248. https://doi.org/10.1016/j.cub.2017.10.026

Tan, S., & Fan, J. (2023). A systematic review of new empirical data on lucid dream induction techniques. Journal of Sleep Research, e13786. https://doi.org/10.1111/jsr.13786

Vallat, R., & Walker, M. P. (2021). An open-source, high-performance tool for automated sleep staging. eLife, 10, e70092. https://doi.org/10.7554/eLife.70092

Walch, O., Huang, Y., Forger, D., & Goldstein, C. (2019). Sleep stage prediction with raw acceleration and photoplethysmography heart rate data derived from a consumer wearable device. Sleep, 42(12), zsz180. https://doi.org/10.1093/sleep/zsz180

Whitmore, N. W., Harris, J. C., Kovach, T., & Paller, K. A. (2022). Improving memory via automated targeted memory reactivation during sleep. Journal of Sleep Research, 31(6), e13731. https://doi.org/10.1111/jsr.13731