Sleep Monitoring with Portable Wi-Fi

Name1 Surname[1,2◑], Name2 Surname[2◑], Name3 Surname[2,3¤], Name4 Surname[2], Name5 Surname[2‡], Name6 Surname[2‡], Name7 Surname[1,2,3*], with the Lorem Ipsum Consortium[¶]

1 Affiliation Dept/Program/Center, Institution Name, City, State, Country
2 Affiliation Dept/Program/Center, Institution Name, City, State, Country
3 Affiliation Dept/Program/Center, Institution Name, City, State, Country

◑These authors contributed equally to this work.
‡These authors also contributed equally to this work.
¤Current Address: Dept/Program/Center, Institution Name, City, State, Country
†Deceased
¶Membership list can be found in the Acknowledgments section.
* correspondingauthor@institute.edu

Abstract

WiFi human sensing is being challenged by researchers continuously e.g. Activity Classification, Localization and etc. However, we believed WiFi can do more. In this paper, we paired WiFi transmitting pattern to human sleep stage to create a mapping rule of those. we used ESP32 which is a reachable single board computer and strengthened its WiFi signal with 2 directional external antennae. After gathering WiFi variation with Sleep Stage from a sleep tracker, we applied machine learning to merge them. We used Long-short term memory (LSTM) to the framework since we know pattern of moving matters to Sleep Stage. So, we are to map a sleeping stage to a sequence of WiFi snapshots instead of one-to-one like others which makes this called "Sleep monitoring". we tested with newly collected data sizing more than 6,000 frames of sequences. To evaluate the result, we compared annotated Sleep Stage with predicted Sleep Stage and summarized their accuracy which finally resulted as that it is possible to classify 4 human sleep stages from WiFi in ESP32.

Introduction

Wi-Fi is one of the most common network mediums nowadays. Pervasively, it is used for establishing a wireless network to connect to the internet. But, there are still many more functions Wi-Fi is good at. Wi-Fi can also be applied in fields beside connecting to the internet according to its stability being upgraded continuously. Decent Wi-Fi connectivity can extract more data other than the data to be transmitted like concentration, speed, obstacle between the transmission. Those can be composed to be many useful applications like Localization, Activity Classification and etc.

In order to achieve the applications like mentioned, There are many works tried to extract deep features from Wi-Fi. But, they are mostly working with very specific tools and Network Interface Card (NIC) connected to a laptop running Linux that is currently one of the ways allowing to obtain fine-grain Channel State Information (CSI), the descriptive data of the Wi-Fi propagating in that environment. Those limitations significantly decrease simplicity of implementation. It is hard for public demonstration and integration with many updated tools.

Actually, there are other existing ways for obtaining the CSI. One is from a ubiquitously used microprocessor, ESP32. which is still not much explored in Wi-Fi exploiting field. It is simple to implement and can be easily integrated with other tools in many platforms due to its massively produced external tools.

Human sleep stage monitoring is being observed for a whlie and recently applied on commercial products. As sleeping is significant to all persons and can lead to many serious deceases, people are more concerned on utilizing their sleepness with using devices those are able to inform them their sleep information e.g. smartwatches, smart mattress, smart rings and etc. But, those devices are considerably affect sleepness since it needs a contact to users' bodies.

So, this paper proposes a contactless sleep monitoring with Wi-Fi CSI from ESP32 where users can applied with easy-to-find microprocessor and monitor their sleep stage without contacting to their bodies. This allows sleep monitoring to be practical for other researchers and those who want to utilize their sleepness.

Background

Human sleep stage monitoring

Human sleep stage monitoring is widely used as stated above. There are many types of staging. But most common and what we chose are stages of 4 which are Wake, REM (Rapid Eye Movement), Light Sleep (Light) and Deep Sleep (Deep). Many commercial devices can log such information in time sequence. We use Fitbit, a popular sleep tracker watch, as our ground truth as it is a device with the most accuracy at Sleep Stage Estimation among others (ref). The Fitbit can log stages with resolution of 30 seconds so, we consider a single Sleep Stage with a sequence of 30-second CSI data from Wi-Fi.

Wi-Fi

Wi-Fi is a well-known connectivity with no wire needed (wireless). It has been used as a medium for connecting to the internet for over 10 years. However, the Wi-Fi is the name covering IEEE 802.11 n/g/ac protocols. It delivers data through 2.4/5GHz frequency with multiple channels. The bandwidth in each channel is 22MHz. the data are to be transmitted parallelly with multiplexing technique named orthogonal frequency division multiplexing (OFDM). Each carrier may propagate to a receiver with encountering many obstacles. The effect of that situation is the Doppler Effect. So, Channel State Information (CSI) is represented as physical layer indicator that can be used to investigate how each channel propagate to the receiver or back to the transmitter.

If a sender sends data to a receiver through Wi-Fi, the data will be almostly not transmitted without any interference.

CSI data

As mentioned in Wi-Fi that data propagating to the receiver while touching surrounding environment, the CSI is a variation of the data. The CSI can be found at both sender and receiver since receiver may transmit data back. Let the sender use the modulation method of 16-quadrature amplitude modulation (16-QAM) which one carrier can carry 4 bits. When the sender needs to send a '1111', the modulation returns $x = 1 + 1i$. Then, transmit to the receiver. At the receiver, let the obtained data is $y = 0.8 + 0.9i$. So, the CSI can be computed by the variation $h = y/x = 0.2 + 3.4i$.

Human body is literally water which reflect radio wave like Wi-Fi. [6], [7] and [8] have proven that human body can affect the CSI.

ESP32

ESP32 is a very popular single-board computer (SBC). With its affordable price and many available additional tools, ESP32 is commonly used in Internet of Things field. Quantitative CSI can be obtained from Wi-Fi in ESP32 according to [26]. The number of available subcarriers in ESP32 is 64.

According to the detail about Wi-Fi mentioned in Wi-Fi, the Wi-Fi in ESP32 has some limitation. It supports only 2.4GHz frequency and can be set only one channel over a connection. The bandwidth of each channel is 22MHz. The CSI can be both obtain from Access point (AP) and Station (STA) as shown in Fig. 1. In this paper, we consider to mainly use CSI at the AP. The frequency of each channel is as 802.11 standard.

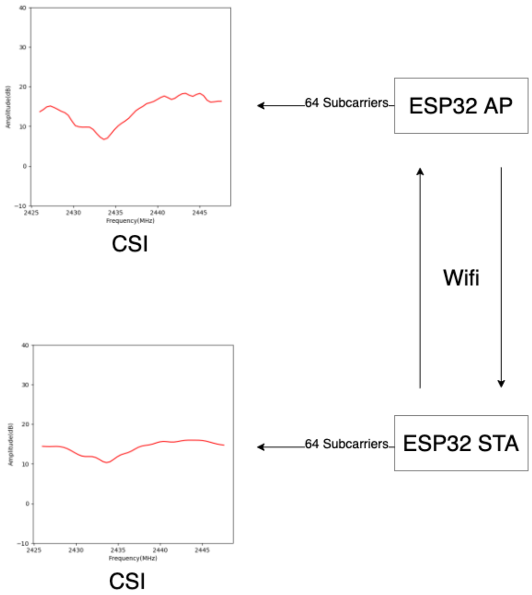

Fig 1. CSI from ESP32s with channel 6.

Materials and methods

Concept

Other famous proposed works like [6] [7] and [25] use omnidirectional antennas while our work uses 2 directional Wi-Fi antennae facing toward human in order to enhance thier precision as shown in Fig. 10.

The CSI is not only affected by human body but mainly by overall environment. This means that 2 identical human poses can result obviously different CSIs if the environment around are not exactly the same as shown in Fig. 2.

So, the definite detection of human standing still in every environment is nearly impossible since the CSI of that situation may be found exactly matched to a CSI of the environment that a big bottle of water placed in front of ESP32. In short, CSI is suitable for moving target since we can focus on its change. The example of mapping

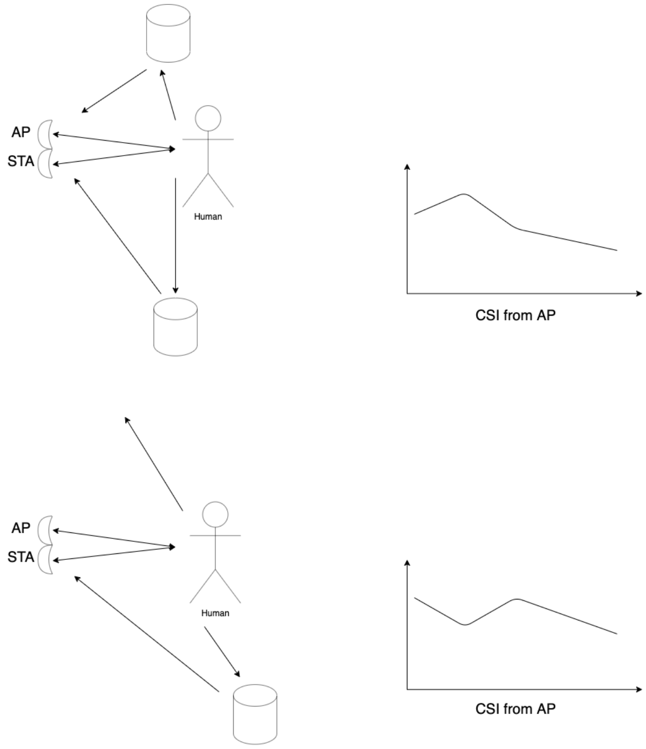

Fig 2. 2 different CSIs resulted from corresponding human poses.

CSI's change to Activity Classification can be found in [8] and [27]. Our work does likewise but focusing on Sleep Stage Monitoring instead of Activity Classification.

In different environment, the CSIs are different. But, the corresponding changing of Sleep Stage may affect to the same changing pattern of CSI. This hypothesis is investigated in the upcoming parts.

All steps of training and testing method are shown in Fig. 3 and Fig. 4 respectively.

Training

CSI Resampling

As mentioned in ESP32, there are 64 subcarriers in CSI data from ESP32 but there are only 52 those are usable while the rest are null. So, we can construst a tensor of 1×52 to represent each CSI. We are to map CSI from the ESP32 to human sleep stage from (Fitbit). The sampling rate of the sleep stage are locked at 30 seconds since the device's limitation. So, we have one sleep stage annotations for 30 seconds. For the ESP32, the sampling rate is originally unpredictable and not constant but it is running around 70Hz. So, we do a process called "Resampling" to the CSI data in order to maintain the dimension of each sequence. The CSI data can be resampled into any size so, we determine it as one of the parameter.

An example of CSI Resampling is shown in Fig. 5. The top graph shows that the the original CSI is logged unstably. The bottom one is to recreate a CSI data at rate 30Hz by calculating each with 2 data at the closet timestamps from the original with a simple mathematical weight equation as in Eq. 1 in order to recreate a CSI data with fixed identical size for all sequences.

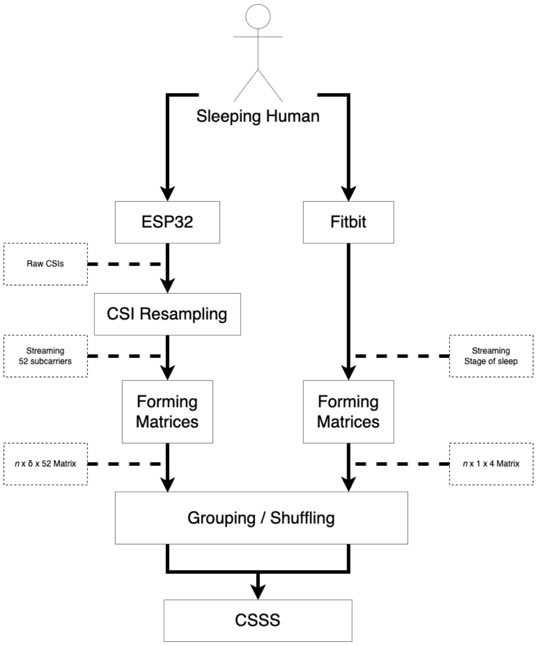

Fig 3. All steps of training method.

$$CSI_{now} = CSI_{before} + \left(\frac{ts_{now} - ts_{before}}{ts_{after} - ts_{before}} \times (CSI_{after} - CSI_{before}) \right), \tag{1}$$

where ts_{now}, ts_{before}, ts_{after} are desired timestamp, timestamp at the closest CSI before the desired timestamp and timestamp at the closest CSI after the desired timestamp respectively. And, CSI_{now}, CSI_{before}, CSI_{after} are CSI at the desired timestamp, CSI before the desired timestamp and CSI after the desired timestamp respectively.

Thus, we are able to map each CSI sequence of 30 seconds to each Sleep Stage annotation.

Data Preparation

Let D be a set of synchronized CSI sequences and Sleep Stage package. Each pair has corresponding timestamp.

$$D = (C_t, S_t), t \in [1, n], \tag{2}$$

where n is a number of pairs, t is a timestamp, C_t is a sequence of CSI data which is collected between $t - 30$ to t and S_t is a Sleep Stage annotation as the ground truth which are collected at t.

All Cs has the same dimension since they are all resampled beforehand. Let δ be the length of each C.

Forming Matrices

To make the CSI and Sleep Stage data valid to the training model, they are to be formed in to a sequence of matrices.

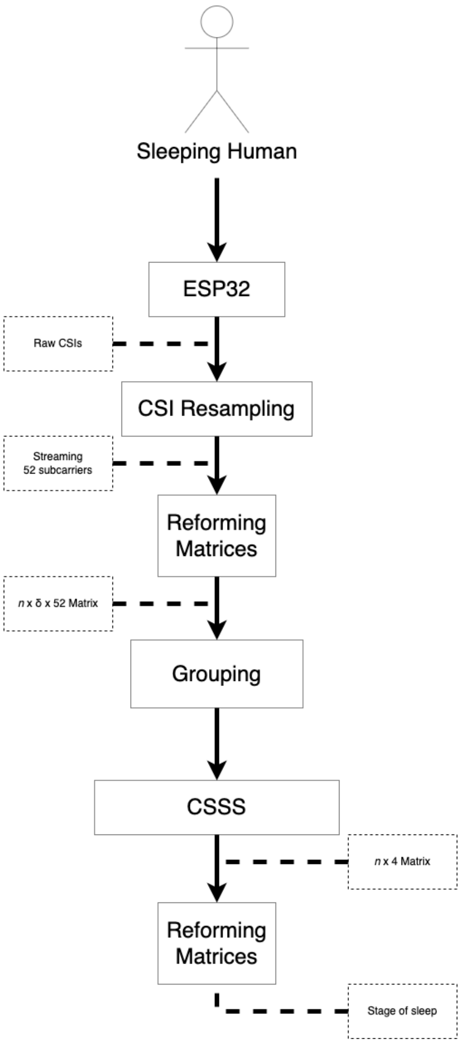

Fig 4. All steps of testing method.

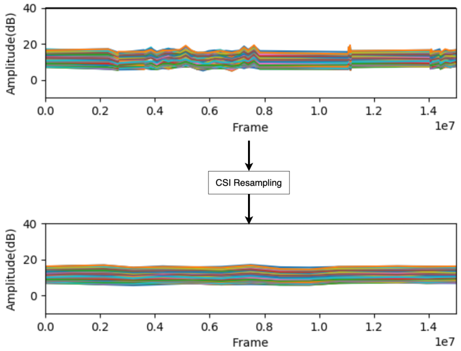

Fig 5. An example of CSI resampling.

CSI data For CSI data (left part in Fig. 3), the streaming data is cut into a serie of $n \times \delta \times 52$ matrices.

However, original CSI data is complex number so, we can either convert it into phase or amplitude. As mentioned in , the CSI is originally in form $h = y/x = v + wi$ so, they are parsed. For amplitude, $\sqrt{v^2 + w^2}$ and $\arctan 2(v, 2)$ for phase. In the paper, we use amplitude.

$$amp = \sqrt{v_{sc}^2 + w_{sc}^2}, sc \in [1, 52] \tag{3}$$

and

$$C = [amp_1, ..., amp_\delta]. \tag{4}$$

The serie length is n so, a single matrix of $n \times \delta \times 52$ is for the CSI data side.

Sleep Stage data For Sleep Stage data (right part in Fig. 3), the streaming data is cut into a serie of n length resulting a single matrix of $n \times 1$. Furthermore, each is mapped into a 1×4 weight matrix as follows.

$$S = \begin{cases} [1, 0, 0, 0] & \text{,if sleep stage is Wake} \\ [0, 1, 0, 0] & \text{,if sleep stage is REM} \\ [0, 0, 1, 0] & \text{,if sleep stage is Light} \\ [0, 0, 0, 1] & \text{,if sleep stage is Deep} \end{cases} \tag{5}$$

Eventually, a single matrix of $n \times 1 \times 4$ is resulted and used for the Sleep Stage data side.

Grouping / Shuffling

The training model swallows training data D as an input, where each is a sequencial set of (C_t, S_t) at a corresponding timestamp with size n. As each C is a sequencial set with size δ, we assume that Long-short term memory (LSTM) [28] is suitable for this type of data.

Moreover, the order of D are shuffled in order to reduce the sequencial relation that can be embedded within the data.

Forming Network Layer

CSI Sequence to Sleep Stage (CSSS) The summation of layers is shown in Table 1. It is designed to shape a CSI Sequence ($n \times \delta \times 52$ tensor) to a predicted Sleep Stage ($n \times 1 \times 4$ tensor).

Table 1. Layers in CSSS.

Layer (type)	Output Shape	Param #
Input Layer	(None, δ, 52)	0
Bidirectional LSTM	(None, δ, 400)	404800
Attention Layer	(None, 400)	160800
TimeDistributed over Dense	(None, 4)	1604

We first use Input Layer with the input size of $(\delta, 52)$ to satisfy dimension of sequential C in D. Then, use Bidirectional LSTM as an encoder layer. The Attention Layer is added afterward to make the model treat with correct time-step. Lastly, we dense the decoder to be 4 where is reshaped into 1×4 latter.

Testing

Most of the processes are similar to those which are in the training method.

Reforming Matrices

Sleep Stage data For Sleep Stage data (in Fig. 4), the data is resulted as a serie of n length resulting a single matrix of $n \times 4$. Each row are reshaped into 1×4 matrix and reverted back with the same manners as the Sleep Stage data parsing in Eq. 5 as follows.

$$\text{The Sleep Stage is} \begin{cases} \text{Wake} & , \operatorname{argmax}_i S'_i = 1 \\ \text{REM} & , \operatorname{argmax}_i S'_i = 2 \\ \text{Light} & , \operatorname{argmax}_i S'_i = 3 \\ \text{Deep} & , \operatorname{argmax}_i S'_i = 4 \end{cases} \tag{6}$$

Results

Data Collection

We recruited 10 volunteers to sleep in between the devices (2 WiFi anntennae) while wearing a Fitbit sleep tracker. The example of the data collection are shown in Fig. 6, Fig. 7, Fig. 8 and Fig. 9 where the top is annotated Sleep State from the Fitbit and the bottom is a resampled WiFi CSI with $\delta = 50$.

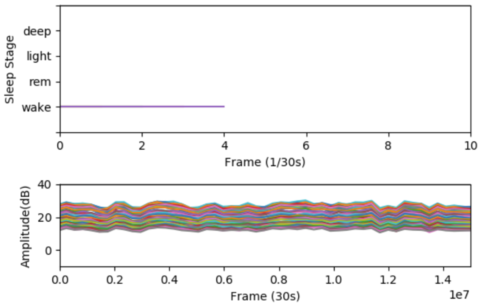

Fig 6. An example of data collection with Wake Stage.

The whole data collection is 50 hours of sleep which worth 6,000 frames for 30 seconds/Sleep State rate. The ratio of dataset for training and testing is 85/15.

Experimental Result

An evaluation is achieved by the algorithm written in Python 3.8. The code is available on Github[1].

As parameters stated in the previous sections, we demostrated the prediction by the followings.

Table 2, Table 3, Table 4 and Table 5 show the estimation performance of 4 Sleep Stages when $\delta = 15$, $\delta = 30$, $\delta = 50$ and $\delta = 200$ respectively.

[1] https://github.com/rtmtree/CSPS

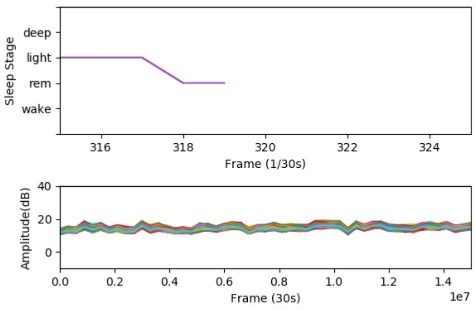

Fig 7. An example of data collection with REM Stage.

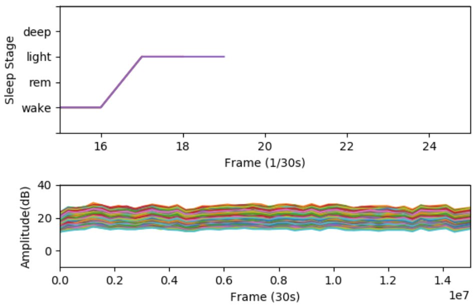

Fig 8. An example of data collection with Light Stage.

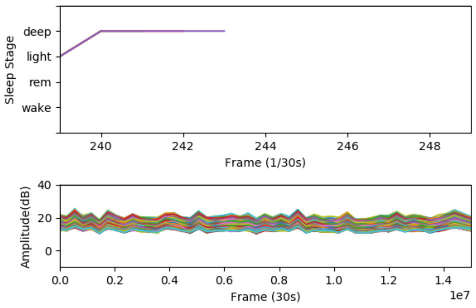

Fig 9. An example of data collection with Deep Stage.

Table 2. Table of evaluation result when $\delta = 15$.

Predicted Truth	Wake	REM	Light	Deep
Wake	0.19	0.06	0.72	0.03
REM	0.03	0.83	0.14	0.00
Light	0.01	0.05	0.91	0.03
Deep	0.00	0.01	0.16	0.83

Table 3. Table of evaluation result when $\delta = 30$.

Predicted Truth	Wake	REM	Light	Deep
Wake	0.28	0.16	0.50	0.06
REM	0.00	0.82	0.18	0.00
Light	0.01	0.04	0.92	0.03
Deep	0.01	0.00	0.20	0.79

Table 4. Table of evaluation result when $\delta = 50$.

Predicted Truth	Wake	REM	Light	Deep
Wake	0.34	0.06	0.5	0.09
REM	0.00	0.83	0.17	0.00
Light	0.01	0.03	0.92	0.04
Deep	0.01	0.00	0.15	0.85

Table 5. Table of evaluation result when $\delta = 200$.

Predicted Truth	Wake	REM	Light	Deep
Wake	0.19	0.06	0.66	0.09
REM	0.03	0.76	0.21	0.00
Light	0.01	0.04	0.91	0.04
Deep	0.01	0.00	0.17	0.82

Discussion

Environment Installation

The Installation of WiFi antennae can affect tremendously in data. We fixed the positions in both training and testing process as shown in Fig. 10. 2 antennae are connecting to ESP32s and to the PC afterward for the AP antennae for CSI collection. Lastly, the human sleeping on the bed need to wear Fitbit sleep tracker as shown in Fig. 11.

CSI Extraction Method

There are many solutions for extracting WiFi CSI from the ESP32 as mentioned in ESP32. This paper picked the solution from ESP32-CSI-Tool[1] since it is considerably well-written and simple to organize. To change the method of extracting WiFi CSI would affect the result significantly.

[1] https://github.com/StevenMHernandez/ESP32-CSI-Tool

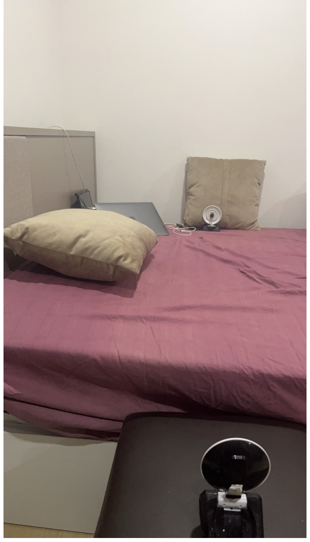

Fig 10. Camera setup and a bed.

Fig 11. Fitbit sleep tracker.

Multiple Person Sleep Stage Estimation

The model does a mapping rule from WiFi CSI with a specific dimension to human sleep stage matrix. So, it is able to only detect single person pose in a range. The annotations originally can result all human sleep stage by using the devices for all specimen. If we do training with multiple sleep stage data instead, we do not believe the result will be well since CSI of a person moving is rarely separatable form others.

Conclusion

Currently, there are many concern on sleepness in all group of people. Especially, those who works in extraordinary time period are facing problem of having deficient sleepness. They are pleased to pay for a device that can inform if their sleepness is well enough e.g. smartwatch, smart ring and smart mattress. Those options come with additional and uncomfortable since users' bodies need a contact to an external device. So, we aim for proposing a contactless Sleep Monitoring device.

After having a research, we discoverd that a variation in WiFi called WiFi CSI can tell whether the area is having an activity or moving objects. Moreoever, [8], [27] and many other works had been done very well on detecting even what kind of activities is happening in the area. We do not believe that this is the limitation of WiFi CSI. In order to solve the above problem and prove if WiFi CSI is precise enough, we tried to overcome this by extracting a deeper information like Sleep Stage Estimation from it.

We controlled environments and enhanced WiFi antennae stability as much as possible then mapped it to the Sleep Stage annotated by Fitbit technology. The result was very poor since the WiFi CSI is very vague. It penetrates through most of the things. We learned from [20] that WiFi CSI value does not matter than its change. The works used Long-short term memory (LSTM), a neural network where focusing on sequence of the data, and obtained a very good result.

We applied the idea of LSTM instead and obtained a lot better result with more than 80% of overall accuracy. Then, we adjusted the model to be suitable for our type of data and did fine-tuning for the frame size to be fit the most for normal human speed of moving while sleeping. So, This paper proposed a model of mapping rule that can takes a sequence of 30-second WiFi circumstance as an input and return an according human sleep stage as an output. The work can help people to monitor Sleep Stage of elderlies and those who are in need of sleep concerning. Moreover as a contact is not needed, users can not differentiate thier sleeping with or without the device. Significantly, the WiFi CSI obtained from ESP32 which is affordable and widely reachable so, people can simply apply the method on their own. The result of various fine-tuned environments and parammeters is acceptable and shown in Experimental Result.

Acknowledgments

We thank to Sirindhorn International Institute of Technology for providing technical environment and supportive information.

References

1. Conant GC, Wolfe KH. Turning a hobby into a job: how duplicated genes find new functions. Nat Rev Genet. 2008 Dec;9(12):938–950.

2. Ohno S. Evolution by gene duplication. London: George Alien & Unwin Ltd. Berlin, Heidelberg and New York: Springer-Verlag.; 1970.

3. Magwire MM, Bayer F, Webster CL, Cao C, Jiggins FM. Successive increases in the resistance of Drosophila to viral infection through a transposon insertion followed by a Duplication. PLoS Genet. 2011 Oct;7(10):e1002337.

4. Osokin D. Real-time 2D Multi-Person Pose Estimation on CPU: Lightweight OpenPose. arXiv:1811.12004 [cs.CV]. 2018 Nov;18.

5. Mehta D, Sotnychenko O, Mueller F, Xu W, Sridhar S, Pons-Moll G, Theobalt C. Single-Shot Multi-Person 3D Pose Estimation From Monocular RGB. arXiv:1712.03453 [cs.CV]. 2018 Aug;28.

6. Wang F, Panev S, Ziyi D, Han J, Huang D. Can WiFi Estimate Person Pose?. arXiv:1904.00277 [cs.CV]. 2019 Apr;2.

7. Liu J, Liu H, Chen Y, Wang Y, Wang C. Wireless Sensing for Human Activity: A Survey. IEEE COMMUNICATIONS SURVEYS & TUTORIALS, VOL. 22, NO. 3, THIRD QUARTER 2020.

8. Chowdhury TZ, Leung C, Miao CY. WiHACS: Leveraging WiFi for Human Activity Classification using OFDM Subcarriers' correlation. IEEE, GlobalSIP 2017.

9. Guo L, Wang L, Liu J, Zhou W, Lu B. HuAc: Human Activity Recognition Using Crowdsourced WiFi Signals and Skeleton Data. Wireless Communications and Mobile Computing Volume 2018, Article ID 6163475.

10. Wang F, Feng J, Zhao Y, Xiaobin Zhang, Zhang S. Joint Activity Recognition and Indoor Localization with WiFi Fingerprints. arXiv:1904.04964 [cs.CV]. 2019 Jul;18.

11. Al-qaness MAA, Li F, Ma X, Zhang Y, Liu G. Device-Free Indoor Activity Recognition System. Appl. Sci. 2016, 6, 329; doi:10.3390.

12. Wang W, Liu AX, Shahzad M, Ling K, Lu S. Device-free Human Activity Recognition Using Commercial WiFi Devices. IEEE Journal on Selected Areas in Communications. DOI 10.1109/JSAC.2017.2679658.

13. Zhao T, Li F, Tian P. A Deep-Learning Method for Device Activity Detection in mMTC Under Imperfect CSI Based on Variational-Autoencoder. IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. 69, NO. 7, JULY 2020.

14. Liu J, Teng G, Hong F. Human Activity Sensing with Wireless Signals: A Survey. Sensors 2020, 20, 1210; doi:10.3390/s20041210.

15. Yousefi S, Narui H, Dayal S, Ermon S, Valaee S. A Survey on Behaviour Recognition Using WiFi Channel State Information. arXiv:1708.07129 [cs.AI]. 2017 Aug;23.

16. Chen Z, Zhang L, Jiang C, Cao Z, Cui W. WiFi CSI Based Passive Human Activity Recognition Using Attention Based BLSTM. IEEE TRANSACTIONS ON MOBILE COMPUTING, VOL. 18, NO. 11, 2019 Nov.

17. Li B, Cui W, Wang W, Zhang L, Chen Z, Wu M. Two-Stream Convolution Augmented Transformer for Human Activity Recognition. Association for the Advancement of Artificial Intelligence, 2021.

18. Luo Y, Ren J, Wang Z, Sun W, Pan J, Liu J, Pan J, Lin L. LSTM Pose Machines. arXiv:1712.06316 [cs.CV].

19. Lee K, Lee I, Lee S. Propagating LSTM: 3D Pose Estimation based on Joint Interdependency. Computer Vision – ECCV 2018. ECCV 2018. Lecture Notes in Computer Science, vol 11211. Springer, Cham.

20. Du X, Vasudevan R, Johnson-Roberson M. Bio-LSTM: A Biomechanically Inspired Recurrent Neural Network for 3D Pedestrian Pose and Gait Prediction. arXiv:1809.03705 [cs.RO]. 2019 Sep;13.

21. Hossain MRI, Little JJ. Exploiting temporal information for 3D human pose estimation. arXiv:1711.08585 [cs.CV]. 2018 Sep;12.

22. Pavllo D, Feichtenhofer C, Grangier D, Auli M. 3D human pose estimation in video with temporal convolutions and semi-supervised training. arXiv:1811.11742 [cs.CV]. 2019 Mar;29.

23. Chen T, Fang C, Shen X, Zhu Y, Chen Z, Luo J. Anatomy-aware 3D Human Pose Estimation with Bone-based Pose Decomposition. arXiv:2002.10322 [cs.CV]. 2021 Jan;26.

24. Ruiz AH, Porzi L, Bulo SR, Moreno-Noguer F. 3D CNNs on Distance Matrices for Human Action Recognition. MM '17, Mountain View, CA, USA. 2017 Oct;23–27.

25. Hernandez SM, Bulut E. Lightweight and Standalone IoT based WiFi Sensing for Active Repositioning and Mobility. IEEE 21st International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM), 2020.

26. Atif M, Muralidharan S, Ko H, Yoo B. Wi-ESP—A tool for CSI-based Device-Free Wi-Fi Sensing (DFWS). Journal of Computational Design and Engineering, 2020, 7(5), 644–656.

27. Zou H, Zhou Y, Yang J, Jiang H, Xie L, Spanos CJ. DeepSense: Device-free Human Activity Recognition via Autoencoder Long-term Recurrent Convolutional Network. 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 2018, pp. 1-6, doi: 10.1109/ICC.2018.8422895.

28. Hochreiter S, Schmidhuber J. LONG SHORT-TERM MEMORY. Neural Computation 9(8):1735-1780, 1997.