IoT System for Sleep Quality Monitoring using Ballistocardiography Sensor

Nico Surantha[1], C'zuko Adiwiputra[2], Oei Kurniawan Utomo[3], Sani Muhamad Isa[4], Benfano Soewito[5]
Binus Graduate Program – Master of Computer Science
Bina Nusantara University, Jakarta, Indonesia

***Abstract*—Sleep is very important for people to preserve their physical and mental health. The development of the ballistocardiography (BCG) sensor enables the possibility of day-to-day and portable monitoring at home. The goal of this study is to develop an IoT sleep quality monitoring system using BCG sensors, microcontrollers and cloud servers. The BCG sensor produces ECG data from the physical activity of the patient. The data is sent to the sensor and is read by the microcontroller. The sensor data is collected and pre-processed in the microcontroller. The microcontroller then transmits the data obtained from the BCG sensor to the cloud server for further analysis, i.e. to assess the sleep quality. The assessment of data transmission efficiency and resource consumption is carried out in this paper. The findings of the evaluation show that the proposed method achieves higher efficiency, lower response time and decreases memory usage by up to 77% compared to the conventional method.**

Keywords—Internet-of-Things; sleep quality; ballisto-cardiography; HRV; ECG

I. Introduction

Sleep is a daily rhythm or circadian rhythm in humans that is governed by the human biological clock in the brain's core hypothalamus [1]. In sleeping conditions, the body will become more relaxed and have minimal movement due to a decrease in attention to the surrounding environment. Sleep is required by humans in order to sustain human health conditions, because it improves the recovery of the exhausted body when it is in the wake, it helps the cognitive function of the brain due to the cleaning of residual waste in the brain [2].

Humans who do not get enough sleep, which is 7-9 hours a day in a certain period of time, result in people experiencing sleep disorders or sleep disorders [3]. Some problems may arise from sleep disorders and disorders such as cardiovascular disease, obesity and diabetes, psychiatric illness, cancer [4] and sleep apnea [5]. Therefore, there is a need for a supporting method or mechanism to track the quality of human sleep to avoid these issues.

One way to monitor the quality of one's sleep is to use the polysomnography (PSG) process. In its calculation, PSG has several criteria, namely the measurement of brain activity, eye movements, breathing and the body. PSG in the medical world is the gold standard for testing the quality of one's sleep, but it has some drawbacks, e.g. the system is only available in every hospital, the procedure is costly, and the sensor is too obstructive in the human body [6].

The use of sensors on the subject will reduce sleep comfort and will certainly have an impact on the quality of sleep measured. The technology developed as the latest state-of - the-art is ballisticocardiography (BCG) sensors. BCG is a heart rate monitor (ECG) and a respiration sensor that works wirelessly, i.e. without interaction with the body of the person being tested [7].

Data from BCG will be used to develop an IoT network consisting of sensors, microcontrollers and web applications on a cloud server. The system is designed to collect patient sleep data by recording directly using a ballistocardiography monitor, the output of which comes in the form of a dataset for heart rate variability (HRV). Those data will then be used to classify sleep cycles in humans, such as waking, light sleep, deep sleep, and rapid eye movement (REM) [8]. Of the 4 sleep stages, the patient's sleep quality will be classified using a combination of the Weight Extreme Learning Machine (WELM) method and the Particle Swarm Optimization (PSO) method used by Utomo et al. (2019) [9] with an accuracy rate of 78.78% in three sleep classes (NREM, REM, awake) and an accuracy rate of 73.09% in four sleep classes.

This scientific study has been carried out in response to previous research which still needs studies in the construction of a portable, high-performance and low-resource-consumption health monitoring system to monitor sleep quality using BCG sensors. The result of this research is expected to provide options for monitoring the quality of human sleep and to provide reports to medical experts on a periodic basis by presenting the results of the extraction of health data from the implementation of IoT. The cost of medical service is therefore cheaper and the patient does not need to see a doctor every time to check the quality of his sleep [8]. In this paper we evaluate the performance of proposed systems in terms of of data transmission and resource consumption aspects.

The remainder of the paper is structured as follows. The background material is explained in Section II of this document. The related works are listed in Section III of this text. The system design and simulation results are explained in Section IV and Section V, respectively. The conclusion is explained in Section VI.

II. Background Material

Some of the background materials as shown in Fig. 1 are discussed in this section.

This research is supported by Directorate General of Strengthening for Research and Development, Ministry of Research, Technology, and Higher Education, Republic of Indonesia.

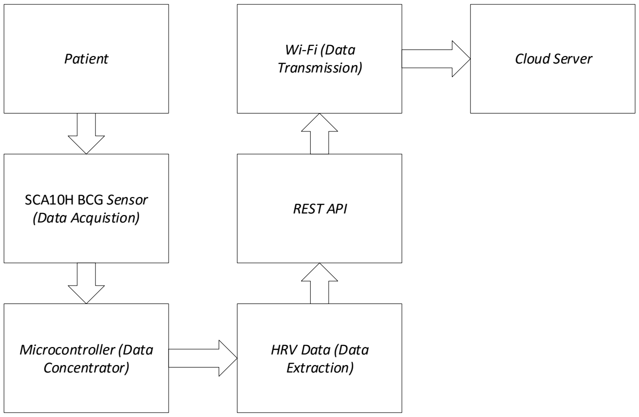
Patient
Wi-Fi (Data Transmission)
Cloud Server
SCA10H BCG Sensor (Data Acquistion)
REST API
Microcontroller (Data Concentrator)
HRV Data (Data Extraction)

Fig. 1. Background material.

A. SCA11H BCG Sensor

SCA11H is a contactless sleep sensor that uses BCG measurements to monitor patients in the hospital and at home. This system allows users to track patients when they are in bed for sleeping conditions [10]. The power needed by SCA10H is 8-10V, 8mA.

B. Data Acquisition

Data Acquisition in this IoT system is performed by measuring physical data in humans, such as ECG, respiration rate, heart activity, and body posture through sensors. The sensor sends data to the cloud server over the network through a data concentrator [11]. The data concentrator can be a smartphone or a microcontroller.

C. Raspberry Pi

Raspberry Pi is a credit card-sized computer with a weight of only 50 grams that can be purchased at a low price compared to the actual computer. Various devices such as mouse, keyboard, Wi-Fi adapter can be connected to four accessible USB 2.0 ports. Raspberry Pi 3 model B + uses a power of 5v, 700mA [12].

D. Data Concentrator

Data Concentrator works to collect data in real time and to send data to data storage in real time, such as a cloud database. Data Concentrator controls communication protocols for sensor data transmission over a certain period of time and interval [13].

E. Data Transmission

Data Transmission is responsible for delivering data from the data concentrator to a data center or cloud device with assured security and privacy. Data transmission is required because, generally speaking, the data acquisition device on the sensor is only equipped with short-range radio technology such as Bluetooth. Data can be sent to long-term data storage via an internet connection in a data concentrater, e.g. via a Wi-Fi network [14].

F. REST

Representational State Transfer (REST) is an architectural design of the web where the architectural strength in the form of a series of constraints is applied to every element contained in the architecture. This makes it easier for a web designer to design a system and impose behavior on the system to be built [15].

III. Related Works

Several methods have been developed to detect portable and wireless heart signals (ECG). Nandakumar et al. (2015) [16] proposes the detection of sleep apnea based on chest and abdomen movements using a smartphone modified to sonar. Smartphone speakers are set to emit waves with a certain frequency and the reflection of the waves is captured by the microphone on the smartphone. The problem is with the non-standard speakers and microphones on each smartphone, and the system is vulnerable to the sound / movement of other body parts besides the chest and abdomen.

The EarlySense sensor proposed by Tal et al. (2017) [17] as a contactless sensor solution for detecting ECG signals The EarlySense sensor must be placed under the bed and will monitor ECG signals while we are sleeping. EarlySense sensor has a problem that is the placement of sensors must be under the bed which will block the bed which will result in sleep becoming uncomfortable. Sensors are also prone to shifting positions that will impact ECG readings.

Araujo et al. (2018) [18] proposes ApneaLink, which is a portable PSG sensor that is used to read data while the user is sleeping and processes the raw data obtained into a sleep apnea analysis. To be able to do the reading, the ApneaLink sensor is bound to the chest and user's stomach. This will reduce the user's comfort while sleeping.

Yacchirema et al. (2018) [19] have developed an IoT device to monitor the quality of health, specifically related to sleep apnea problems. There are three layers proposed by Yacchirema et al. (2018). They are the IoT layer consisting of sensors, the fog computing layer consisting of a data aggregator, and the cloud layer consisting of a database and user interface. The technology used is HDFS, Apache Spark, Kafka, and Cassandra.

Utomo et al. (2019) have carried out research on the IoT platform to analyze the sleep classification or stages of sleep [9]. The ECG signals are selected as data (input) because the ECG sensor is more simple to use. Utomo et al. (2019) have compared several methods for analyzing sleep cycles, including Extreme Learning Machine (ELM), Backpropagation Neural Network (BPNN), and Support Vector Machine (SVM) designed by Lesmana et al. (2018) [20] and proposed to used Weight Extreme Learning Machine (WELM) with combination with Particle Swarm Optimization to counter the imbalance dataset between sleep classes in sleep sensor.

The SCA11H sensor is a BCG sensor developed by the Murata Electronics company [21]. SCA11H is a BCG (contactless) sensor that can be used to measure ECG signals, which has a correlation with PSG for heart rate signal measurements of 0.97 and has an average error of -0.1 ± 4.4 beats per minute for a 95% confidence level [22]. SCA11H sensor has the advantage that it can be installed at the bedside and read data can be sent via a microcontroller connected to a Wi-Fi module.

From previous research, there is a need for portable and regular sleep monitoring system. Many sensor technologies have been explored for this purpose, e.g. mobile phone, wearable sensor, and portable ECG sensor. The emergence of BCG sensor can introduce the possibility of portable and daily monitoring at home. Because BCG is mounted on the bedside, it is supposed to be functional and comfortable for the patient. Therefore, the sleep monitoring system based on BCG sensor is proposed in this research.

IV. System Design

The main objective of the system design is to introduce an IoT system that focuses only on data acquisition, data concentration and data transmission. Development in cloud server and monitoring application has already been done by Utomo et al. (2019) [23]. This system is being reused in this research with some rework and simplification of some services to increase the performance of resources such as memory, CPU, and disk used on a cloud server. The complete architecture of the proposed system is shown in Fig. 2.

A. Data Acquisition

The data acquisition section consists of a SCA11H sensor which is a contactless BCG sensor. The SCA11H sensor has the ability to record heart and respiratory signals. Data read by the sensor will be sent using Wi-Fi (IEEE 802.11). The SCA11H sensor requires power of 9V-8mA and is installed at the bedside to operate. This data acquisition is expected to produce heart rate data to be sent to the data concentrator.

BCG sensor generates data stored on the microcontroller in the form of a text file printed by the Python program and stored also on a cloud server database using the PostgreSQL database. BCG SCA11h sensor variable output is ordered as follow: timestamp, HR, RR, SV, HRV, SS, status, B2B,

B2B1, B2B2. The explanation of each output is explained in Table 1.

Table 1 shows that the data obtained from SCA11H has a lot of payload output with each type and unit. Data is written in one text file, separated by commas between variables for output data between variables and separated by enter to separate data every second.

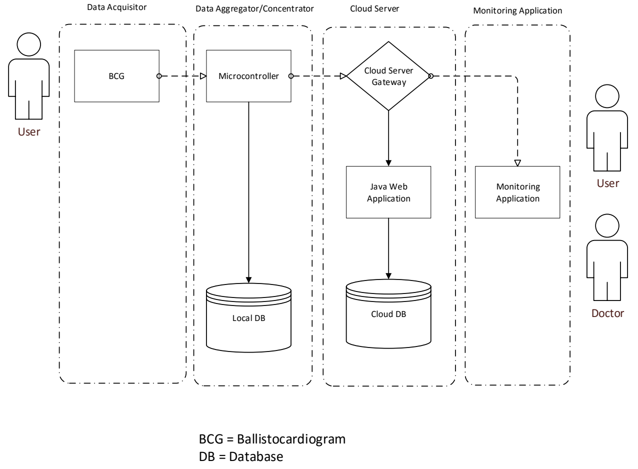

Fig. 2. System Design.

TABLE. I. BCG Sensor Output Data

Variable	Data Type	Unit	Description
time_stamp	S32		System Timestamp
HR	S32	1/min	Heart Rate
RR	S32	1/min	Respiration Rate
SV	S32	ml	Relative Stroke Volume
HRV	S32	ms	Heart rate variability
signal_strength	S32	arbitary unit	measured signal strength indication
status	S32	ms	0 = low signal 1 = ok signal 2 = high signal 3 = (close to) overload 4 = (close to) max HR
B2B	S32	ms	Beat-to-beat time
B2B1	S32	ms	Beat-to-beat time
B2B2	S32	ms	Beat-to-beat time

BCG SCA11H is configured using 1kHz acceleration intervals and runs an algorithm designed by Murata Electronic. Data from the results of the preprocessing algorithm are sent in the output data @ 1Hz rate, which is one data per second. Data at a rate of 1 Hz is also applied when the data representation is issued in the form of a log file in text format.

The study only uses data B2B, B2B1, B2B2 with integer data type type S32. Millisecond value in the process of the unit will be changed to second. This change was made because the preprocessing data made on the cloud server only needs to use B2B data in second units as a reference to classify sleep quality.

B. Data Concentrator

Data concentrator is implemented on Raspberry Pi 3 model B + board. Data from the SCA11H sensor will be sent to the data concentrator, in the form of an endpoint that consists of:

1) Local gateway, which has the function to process raw data storage into a database / database. In addition, the local gateway also functions to order the cloud to perform classification stages of sleep operations.

2) Database to store raw data in the form of text, which has the function to store data from the BCG sensor.

In the data concentrator, there are three pre-process that are applied before transmitting data to the cloud:

1) Data cleansing: is carried out in the data extraction process due to irrelevant ECG data records. To find out the occupancy of the patient's bed, this study discarded the status variable 0 where the status 0 means that the patient's heart rate was not found on the bed that has been installed with BCG SCA11H sensor.

2) B2B data extraction: From all the sensor output as defined by Table 1, only the last 3 output are utilized in this research, i.e. B2B, B2B1, B2B2.

3) B2B data filtering: At this stage if the data obtained has B2B data > 3000 milliseconds, then the data can be removed because this is noise [22]. Furthermore, it also removes unused beat-to-beat empty slots, *i.e.* beat-to-beat which has a value of 0 based on a separate comma that separates data reading. The units in B2B are still in the form of milliseconds, so they need to be changed into second before being sent to the cloud server. The number of B2B samples before and after filtering is shown by Table 2.

TABLE. II. SAMPLE OF BEAT-TO-BEAT (B2B) DATA

Data	**Number of B2B sample (previous)**	**Number of removed B2B sample**	**Number of B2B sample (after removal)**
DemoData1.txt	6741	4034	2707
DemoData2.txt	25602	17413	8189
DemoData3.txt	70224	55159	15065

C. *Cloud Server*

The cloud server section has a function to collect daily sleep quality data from all existing patients as a doctor's reference. On the cloud server the classification process of sleep stages and the level of sleep quality will be carried out. Cloud server will send data that will be displayed on the monitoring application dashboard.

D. *Monitoring Application*

The outcomes of patient sleep monitoring will be shown in a tracking report that can be accessed by patients and health professionals through a web application that can be accessed via a smartphone or a laptop. Information to be obtained by patients and health practitioners at this level may be in the form of tables and graphs that provide information on the quality of sleep monitoring of patients in a specific time period. In this study, the monitoring application is developed using Java Application using the JHipster framework where there is an HTML, CSS, and AngularJS framework as a front-end framework and Java version 1.8.0 181 is used as a back-end application support.

E. *System Integration*

The BCG sensor transmits data on the heart rate to the raspberry pi, which serves as a data focus. The system aggregator / concentrator data includes a program that consists of a python library and a source code. The data concentrator is capable of reading data from the sensor every second with this device. The Python source code named SleepMonitoring. Py will run automatically every time the Raspberry Pi hardware is switched on. The python program can perform pre-processing of heart rate data in the form of data selection, data extraction, and data cleansing.

Data that has been pre-processed will be sent directly to the web application hosted on the cloud server using the RESTful Web Services post form. Patient sleep data will be shown in real-time in the monitoring program. If the patient has finished sleeping, the program on the web server will conduct a quantification process to determine the patient's sleep and sleep quality.

This study uses a service that was developed using Python programming in Raspberry Pi. For all hardware and software used are:

- Microcontroller: Raspberry Pi 3 model b +
- Python version 3
- Raspbian OS version 4.19
- Google Cloud Platform
- SCA10H sensor
- USB to serial TTL Converter PL 2303 YP-01
- Battery 9v as a power supply for the SCA10H sensor
- Charger Adapter for Raspberry Pi
- 2.4GHz and 5 GHz 802.11 ac Wireless LAN on Raspberry Pi 3 model 3+ as a raspberry pi connection to the internet.

V. RESULTS AND DISCUSSION

In this study, we conducted an examination of 26-year-old patients with BCG sensors for three consecutive days that obtained HRV data generating data with the names DemoData1.txt, DemoData2.txt, and DemoData3.txt. This data is stored on the microcontroller in the form of a text file printed by the Python program and stored also on a cloud server database using the PostgreSQL database. In this section, two kinds of evaluation are performed, i.e. data transmission evaluation and resources consumption.

A. *Data Transmission Evaluation*

Previous study, Utomo et al. (2019) [23] uses simulation data in the form of S32 integer HRV data with the second unit value. It also uses the timestamp in the request body for the REST payload post. However, it is not reliable due to the delay when data transmission is performed. Delays in the REST Architecture cause HRV data out of sync when compared to the input time in the simulator data with the data receive time on the cloud server. Therefore, in this research, we use time_stamp and HRV data directly obtained from the SCA10H sensor in real time.

Data transmission and response time testing samples are obtained from Table 1 where the final B2B sample number is sent every 1 second. The error rate parameter in Table 3 is determined by where the error rate is 0%. This result means that all responses received by the microcontroller from the cloud server are HTTP Status 200 OK. In Table 3, performance measurements are carried out to determine the performance of data transmission (average response time, average sent bandwidth, and average received bandwidth). In this evaluation also obtained a delay on the REST architecture is approximately 9-10 ms.

In order to evaluate the amount of transactions that can be received on a cloud server in one second, then testing is carried out to send 1000 data with ramp-up 0 seconds. The data tested are DemoData1.txt, DemoData2.txt, DemoData3.txt. Each data will be tested 20 times to get the average throughput for this implementation. The result of throughput is shown by Fig. 3.

The average of throughput for proposed method is approximately 913 transactions/second which is higher than the conventional method [23]. While, the response time of the proposed method is 224.7 ms which is lower than the conventional method [23]. The comparison of throughput and response time is shown by Table 4.

B. *Resources Consumption Evaluation*

In previous study, the monolith applications designed by Utomo et al. (2019) [23] consumes very large memory that is equal to 1,437.67MB for the use of 1 instance server. Therefore, this study tries to reuse previous research and simplify services built on the Java Application that was built on previous research.

To reduce the burden of performance on the cloud server, this study also modified the Tomcat JDBC database connection pool into the Hikari Connection Pool (HikariCP) database connection pool used in previous study. As shown by Fig. 4, Fig. 5 and Fig. 6, the CPU utilization, storage disk utilization, and RAM consumption are 0.016 mCPU Core, 4.592 MB, and 690.2 MB, respectively. From Table 5, it can also be seen that the proposed method can reduce the memory consumption until 77% from the conventional method [23].

TABLE. III. DATA TRANSMISSION EVALUATION

Parameter	DemoData 1.txt	DemoData 2.txt	DemoData3 .txt
HTTP Request from microcontroller	2707	8189	15065
Evaluation Duration	45 Min 7 Sec	2 H 16 Min 29 Sec	4 Hour 11 Min 5 Sec
Error Rate	0%	0%	0%
Average Response From Cloud	9ms	10ms	9ms
Average Received Bandwidth	0.25KB/sec	0.26KB/sec	0.25KB/sec
Average Sent Bandwidth	0.28KB/sec	0.29KB/sec	0.27KB/sec

TABLE. IV. THROUGHPUT AND RESPONSE TIME COMPARISON

Method	Throughput (transactions/s)	Response Time(ms)
Proposed Method	913	224,7
Conventional Method [23]	803,24	253,34

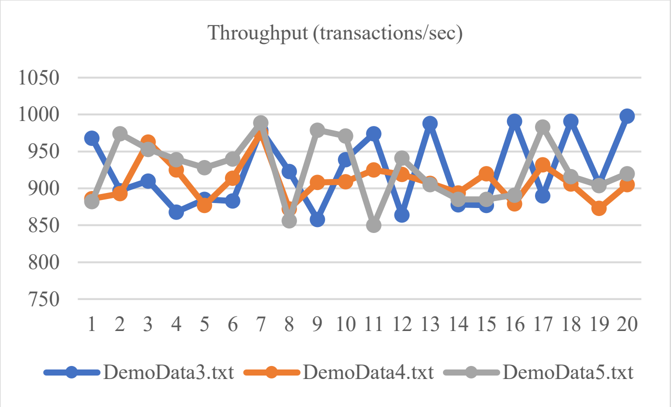

Fig. 3. Throughput Result with RAMP-up 0s.

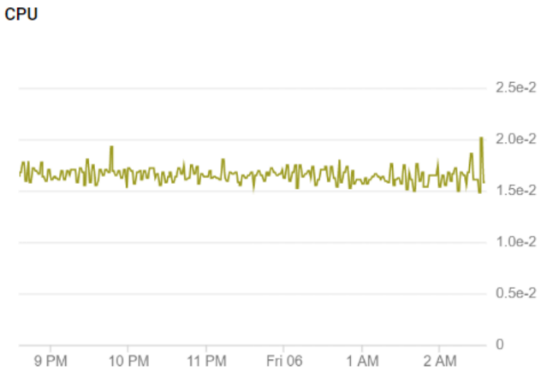

Fig. 4. CPU Utilization During Monitoring.

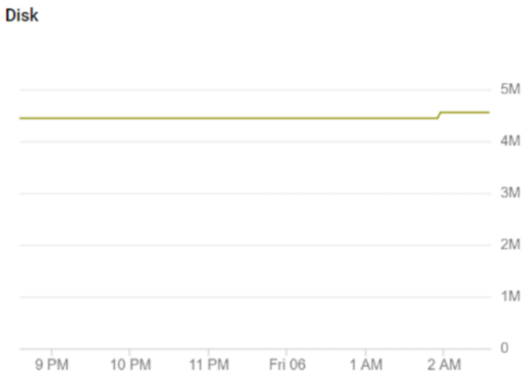

Fig. 5. Storage Consumption.

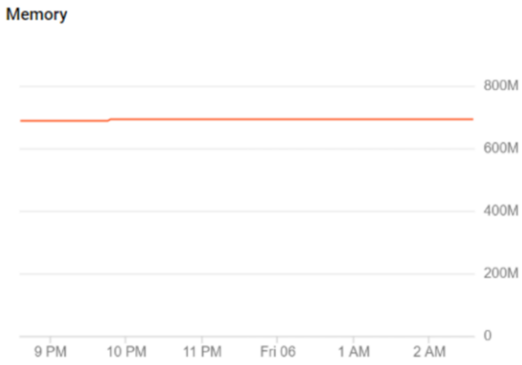

Fig. 6. RAM Consumption.

TABLE. V. RESOURCES CONSUMPTION COMPARISON

Resources Name	Proposed Method	Conventional Method [23]
CPU	0.016 mCPU Core	-
Memory	690.2 MB	2898 MB
Disk	4.592 MB	-

VI. CONCLUSION

In this study, we have proposed the IoT system for sleep quality monitoring using BCG sensor. The utilization of BCG sensor introduces the possibility of daily and portable monitoring of sleep quality at home. We have evaluated the performance of the system in terms of data transmission capacity and resources consumption. The evaluation results that the proposed method can achieve higher throughput, lower response time, and lower memory consumption compared to the conventional method. For the future study, it is recommended to use other extracted sensor data, which includes heart rate, respiration rate, stroke volume, heart rate, variability and beat-to-beat-time as a reference for data to determine a person's sleep quality with more parameters and algorithms complexity.

ACKNOWLEDGMENT

This work is supported by the Directorate General of Strengthening for Research and Development, Ministry of Research, Technology, and Higher Education, Republic of Indonesia as a part of Penelitian Terapan Unggulan Perguruan Tinggi Research Grant to Binus University entitled “Prototipe dan Aplikasi Monitoring Kualitas Tidur Portabel berbasis Teknologi Cloud Computing dan Machine Learning” or “Portable Sleep Quality Monitoring Prototype and Application based on Cloud Computing Technology and Machine Learning” with contract number: 039/VR.RTT/IV/2019 and contract date: 29 April 2019.

REFERENCES

[1] R. E. Mistlberger, B. M. Bergmann, W. Waldenar, and A. Rechtschaffen, “Recovery sleep following sleep deprivation in intact and suprachiasmatic nuclei-lesioned rats,” Sleep, vol. 6, no. 3, pp. 217–233, 1983.

[2] W. H. Moorcroft and P. Belcher, Understanding sleep and dreaming. Springer, 2003.

[3] C. C. Panel et al., “Joint consensus statement of the American Academy of Sleep Medicine and Sleep Research Society on the recommended amount of sleep for a healthy adult: methodology and discussion,” Sleep, vol. 38, no. 8, pp. 1161–1183, 2015.

[4] A. D. Laposky, E. Van Cauter, and A V Diez-Roux, “Reducing health disparities: the role of sleep deficiency and sleep disorders,” Sleep Med., vol. 18, pp. 3–6, 2016.

[5] B. Sulistyo, N. Surantha, and S. M. Isa, “Sleep Apnea Identification using HRV Features of ECG Signals,” Int. J. Electr. Comput. Eng., vol. 8, no. 6, 2018.

[6] A. T. M. VAN DE WATER, A. Holmes, and D. A. Hurley, “Objective measurements of sleep for non-laboratory settings as alternatives to polysomnography--a systematic review,” J. Sleep Res., vol. 20, no. 1pt2, pp. 183–200, 2011.

[7] O. T. Inan et al., “Ballistocardiography and seismocardiography: a review of recent advances.,” IEEE J. Biomed. Heal. Informatics, vol. 19, no. 4, pp. 1414–1427, 2015.

[8] N. Surantha, G. P. Kusuma, and S. M. Isa, “Internet of things for sleep quality monitoring system: A survey,” in Proceedings - 11th 2016 International Conference on Knowledge, Information and Creativity Support Systems, KICSS 2016, 2017.

[9] O. K. Utomo, N. Surantha, S. M. Isa, and B. Soewito, “Automatic Sleep Stage Classification using Weighted ELM and PSO on Imbalanced Data from Single Lead ECG,” Procedia Comput. Sci., vol. 157, pp. 321–328, 2019.

[10] M. Takano and A. Ueno, “Noncontact In-Bed Measurements of Physiological and Behavioral Signals Using an Integrated Fabric-Sheet Sensing Scheme,” IEEE J. Biomed. Heal. Informatics, vol. 23, no. 2, pp. 618–630, 2019.

[11] M. Hassanalieragh et al., “Health monitoring and management using Internet-of-Things (IoT) sensing with cloud-based processing: Opportunities and challenges,” in Services Computing (SCC), 2015 IEEE International Conference on, 2015, pp. 285–292.

[12] M. S. D. Gupta, V. Patchava, and V. Menezes, “Healthcare based on iot using raspberry pi,” in 2015 International Conference on Green Computing and Internet of Things (ICGCIoT), 2015, pp. 796–799.

[13] H. Gabbar, Smart Energy Grid Engineering. Academic Press, 2016.

[14] P. Tomar and G. Kaur, Examining cloud computing technologies through the internet of things. IGI Global, 2017.

[15] M. Hemdi and R. Deters, “Using REST based protocol to enable ABAC within IoT systems,” in 2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2016, pp. 1–7.

[16] R. Nandakumar, S. Gollakota, and N. Watson, “Contactless Sleep Apnea Detection on Smartphones,” in Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services - MobiSys ’15, 2015.

[17] A. Tal, Z. Shinar, D. Shaki, S. Codish, and A. Goldbart, “Validation of contact-free sleep monitoring device with comparison to polysomnography,” J. Clin. Sleep Med., 2017.

[18] I. Araújo et al., “Diagnosis of sleep apnea in patients with stable chronic heart failure using a portable sleep test diagnostic device,” Sleep Breath., pp. 1–7, 2018.

[19] D. C. Yacchirema, D. Sarabia-Jacome, C. E. Palau, and M. Esteve, “A Smart System for Sleep Monitoring by Integrating IoT With Big Data Analytics,” IEEE Access, vol. 6, pp. 35988–36001, 2018.

[20] T. Fennia Lesmana, S. Muhamad Isa, and N. Surantha, “Sleep Stage Identification using The Combination of ELM and PSO based on ECG Signal and HRV,” 2018.

[21] U. Meriheinä, “Monitoring of sleep phenomena.” Google Patents, 2017.

[22] S. Nurmi, T. Saaresranta, T. Koivisto, U. Meriheinä, and L. Palva, “Validation of an Accelerometer Based BCG Method for Sleep Analysis,” Aalto Univ. Publ. Ser. Sci. + Technol., 2016.

[23] O. K. Utomo, “IoT Platform Architecture for Sleep Quality Monitoring,” Bina Nusantara University, 2019.