IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 4, 2022, accepted April 5, 2022, date of publication April 18, 2022, date of current version April 26, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3167637

Intelligent Sleep Monitoring System Based on Microservices and Event-Driven Architecture

NICO SURANTHA[1,2], (Member, IEEE), OEI K. UTOMO[1], EARLICHA M. LIONEL[1], ISABELLA D. GOZALI[1], AND SANI M. ISA[1], (Member, IEEE)

[1]Computer Science Department, BINUS Graduate Program—Master of Computer Science, Bina Nusantara University, Jakarta 11480, Indonesia
[2]Department of Electrical, Electronic and Communication Engineering, Faculty of Engineering, Tokyo City University, Setagaya-ku, Tokyo 158-8557, Japan

Corresponding author: Nico Surantha (nico.surantha@binus.ac.id)

This work was supported by the Directorate General of Higher Education, Ministry of Education, Culture, Research and Technology, Republic of Indonesia, as the part of Penelitian Dasar Unggulan Perguruan Tinggi Research Grant to Binus University entitled "Sistem dan Aplikasi Portabel Pendeteksi Gangguan Irama Jantung Menggunakan Sinyal Electrocardiogram (ECG) Berbasis Machine Learning dan Cloud Computing" or "Portable ECG-Based Heart Arrhythmia Prediction System and Application Using Machine Learning and Cloud Computing" under Contract 163/E4.1/AK.04.PT/2021.

ABSTRACT Sleep monitoring using polysomnography (PSG) in hospitals can be considered expensive, so the preferable way is to use contactless and wearable sensors to monitor sleep daily by patients at home. In this study, the Internet-of-Things (IoT) platform was utilized for sleep monitoring with contactless or wearable sensors as an integrated system developed based on an event-driven and microservice architecture. Multiple services that respond to events are provided within the system. Electrocardiogram (ECG) data were used as the input in the sleep monitoring system. The combination of the weighted extreme learning machine (WELM) algorithm with particle swarm optimization (PSO) was used to process the ECG data, followed by fuzzy logic to measure sleep quality, then display the data on the dashboard. Based on the experimental results, the proposed architecture increased throughput by 34.76%, decreased response time by 55.85%, and reduced memory consumption by 37.26% per instance replication compared to the non-event-driven architecture. The accuracies of the sleep stage classification were 78.78% and 73.09% for the three and four classes, respectively, and the area under a receiver operating characteristic (ROC) curve (AUC) reached 0.89 for both the three and four class classifications.

INDEX TERMS Event-driven architecture, extreme learning machine, Internet of Things, microservices, sleep monitoring.

I. INTRODUCTION

A behavioral state of low awareness or consciousness of the environment can be defined as sleep, in which muscles enter a relaxed state while the human nervous system becomes inactive [1]. Sleep is essential for restoring the body and mind to their original state. Physical and psychological problems such as dizziness and work accidents may occur because of poor sleep quality [2]. As a solution, a sleep monitoring system can be used for the early detection of sleep disorders.

Integrating several technologies, such as sensors, cloud computing, and mobile technology is required for the Internet of things (IoT) in sleep monitoring systems. IoT connects humans with devices and/or digital services, providing a platform for sensor, data processing, and dashboard integration. Hence, sleep data can be easily accessed by the corresponding personnel to monitor and analyze the sleep quality of patients.

Sleep monitoring systems can be designed using several types of architecture [2]–[5]. Typically, three main parts are deployed in the IoT architecture for sleep monitoring: 1) a sensor gateway to gather data, 2) an IoT platform to save and handle data from sleep quality sensors, and 3) dashboards for patients and health experts.

Two concepts can be used to improve the performance of IoT backend services in software architecture technology: event-driven and microservices. In the microservices concept, the system is divided into several independent microservices defined by each function. Microservices consist of components that can be developed and deployed individually using low inter-service coupling to improve the software development process and increase the resource usage efficiency of the servers [6]. In an event-driven concept, events trigger, determine, and build the flow of the system [7].

The associate editor coordinating the review of this manuscript and approving it for publication was Shadi Alawneh.

Event consumers and producers can be decoupled using an event-driven architecture. Communication between producers and consumers, such as routing messages, maintaining topics, and validating messages, can be facilitated by a message broker [8], [9]. Asynchronous processing occurs in an event-driven architecture where a direct connection between the producer and consumer is not possible, resulting in dependency reduction between services and functional capability maximization of each service.

The quantification system for sleep quality is a part of the proposed sleep monitoring system. In 1989, Buyese *et al* [10] determined sleep quality using the Pittsburgh Sleep Quality Index (PSQI) via sleep quality questionnaires based on one-month sleep experience. In 2017, Ang *et al*. [11] used a fuzzy system to quantify sleep quality based on deep sleep, total sleep, and wake duration during a one-night sleep. Total of 27 sets of rules were used to define nine levels of sleep quality.

A classification system is required to classify the sleep stages. Several physiological signals from sensors such as electrocardiography (ECG), electroencephalography (EEG), electromyography (EMG), and electrooculography (EOG) are commonly used by researchers. ECG signals are preferable for sleep studies because they can provide accurate sleep stage classification with simplicity [12]–[17]. Support vector machines (SVMs), Bayesian networks, and deep learning algorithms have been deployed for classification using ECG signals. Based on Moorcroft [1], sleep stages are divided into unequal proportions or imbalanced distributions, resulting in inaccurate classifications. Class distributions of 30.5%, 6.89%, 1.78%, 4.76%, 38.28%, and 17.79% for awake, rapid eye movement (REM), non-rapid eye movement (NREM)4, NREM3, NREM2, and NREM1 stages, respectively, were included in the MIT-BIH polysomnographic database used in this study. A weighted extreme learning machine (WELM) can solve imbalanced dataset problem[18]. WELM is an extreme learning machine (ELM) algorithm with a weight matrix for strengthening the minority class and weakening the majority class.

The incremental contributions in this research are:

1) IoT architecture with applied microservices and event-driven concepts.
2) The combination of WELM with particle swarm optimization (PSO) algorithm was used to classify the sleep stage.
3) Mobile-based dashboard for personal sleep monitoring to display sleep quality data for use by health practitioners.

Good performance (as in better classification accuracy, area under a receiver operating characteristic (ROC) curve (AUC), throughput, memory allocation, and response time) is expected using the novel methods described above. This paper is an extended version of the conference proceedings published in [19].

The remainder of this paper is structured as follows: Section 2 includes the works related to sleep monitoring systems. Section 3 contains information on the proposed IoT platform and the sleep stage classification algorithm, whereas Section 4 presents the proposed sleep-monitoring dashboard. Section 5 provides the performance evaluation of the proposed system, and finally, Section 6 presents the conclusions and future work to complete the paper.

II. RELATED WORK

Software architecture, sleep stage classification, and sleep quality quantification were included in this study. Hence, related work is outlined.

A. THE ARCHITECTURE OF SLEEP MONITORING SYSTEM

Surantha *et al.* [2] surveyed the design of IoT-based sleep monitoring consisting of several parts: 1) data acquisition from sensors, 2) a data aggregator transmitting data to the cloud server, 3) cloud processing for data processing, and 4) an analytics dashboard for application-based monitoring.

In 2016, Kim *et al.* [3] used cloud-based sensing (SenseCloud) as a service (CSaaS), as shown in Fig. 1. Using SenseCloud, users can register sensors, and the sensor data can be stored in SenseCloud storage.

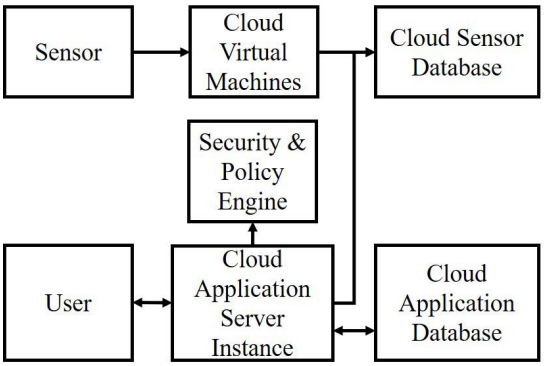


FIGURE 1. Kim et al. [3] architecture of SenseCloud.

In SenseCloud, each service is dedicated to a specific function. The sensor gateway that received data from the sensor was operated using cloud virtual machines and stored in the sensor database. The architecture in this study is denoted as the microservices database (McsDB).

An IoT-based smart healthcare system collecting various sensor data used by health practitioners was developed in 2019 [4], [5] using the architecture shown in Fig. 2. Multiple functions are served by the cloud application, such as 1) obtaining data and information from the sensor gateway, 2) saving the data to the database, and 3) performing data processing. This smart healthcare architecture is denoted as the monolithic architecture (MnlDB) in this work.

B. THE QUANTIFICATION OF SLEEP QUALITY

Ang *et al.* [11] used a fuzzy logic algorithm to quantify sleep quality, taking the percentage of total sleep, deep sleep, and awake duration as input, then categorizing it into bad, good, and sufficient. Three categories were quantified into nine levels of sleep quality, from the worst (L1) to best (L9) using

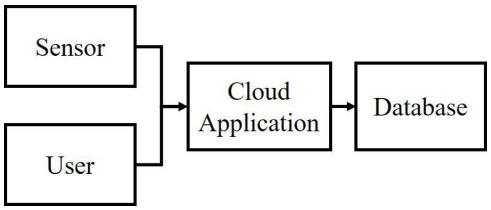


FIGURE 2. Swaroop et al. [4] and Mohaptra et al. [5] architecture of Smart Healthcare.

several sets of rules, in which this study was also implemented using the jFuzzyLogic Java library [20].

C. THE CLASSIFICATION OF SLEEP STAGE

Before quantifying sleep quality, Ang *et al.* [11] used ECG data as inputs to acquire sleep stages. The classification of sleep stages has been conducted using several algorithms.

Adnane *et al.* [12] classified sleep stages into two classes: awake and sleep, with an accuracy of 79.99% using the SVM method. The MIT-BIH polysomnographic database was used as a dataset with ten extracted features. Xiao *et al.* [13] used a random forest algorithm to classify three classes of sleep stages: REM, awake, and NREM. An accuracy of 88.67% was achieved using 41 features.

Another approach was proposed by Fonseca [14] in 2015 using the Bayesian Network multi-class method to classify PSG data into three classes and four classes of sleep stages: awake, REM, deep sleep, and light sleep. For three and four classes, the obtained accuracies were 80% and 69%, respectively, using 80 features.

In 2016, Lesmana *et al.* [15] classified sleep stages using a combination of ELM and PSO algorithms. A total of 18 features from the MIT-BIH polysomnographic database were utilized and achieved accuracy rates of 62.66%, 71.52%, 76.77%, and 82.1% for six, four, three, and two classes, respectively. The accuracy rate of 62.66%, 71.52%, 76.77%, 82.1% was obtained for 6, 4, 3, 2 classes. PSO was used for feature selection, and the ELM algorithm was used to calculate the fitness value. Yücelbaş [16] deployed a morphological method to classify 3 classes of sleep stages using 15 features. The MIT-BIH polysomnographic databases and polysomnography (PSG) were used as datasets with 77.02% accuracy. Wei *et al.* [17] used a deep neural network to classify the three classes using the MIT-BIH polysomnographic database data. An accuracy of 77% was obtained for the 11 features. Table 1 summarizes the related work on sleep classification algorithms.

In the research conducted by Lesmana *et al.* [15], 70% of the dataset was used for training and the rest for testing, using 18 features extracted from ECG data every 30 s. The features are divided into 11 time-domain features and 7 frequency-domain features [16]. Frequency-domain features consist of 1) Total power, 2) low-frequency range (0.04 Hz to 0.15 Hz) (LF), 3) total power in the very-low-frequency range (≤0.04 Hz) (VLF), 4) high-frequency range (0.15 Hz to 0.4 Hz) (HF), 5) LF and HF power in normalized units, and 6) ratio of LF and HF. Meanwhile, the 11 features of time-domain [21] consist of the following: 1) average of RR intervals (AVNN), 2) standard deviation of RR intervals (SDNN), 3) root mean square of the differences of adjacent RR intervals (RMSSD), 4) standard deviation of differences between adjacent RR intervals (SDSD), 5) count of successive differences of RR intervals for more than 50 ms (NN50), 6) the division of NN50 and the total of RR intervals minus 100 in percentage (pNN50), 7) HRV Triangular Index, 8) standard deviation of points which is perpendiculars to the axis of line of identity (SD1), 9) along the axis of line of identity (SD2), 10) ratio of SD1 and SD2, and 11) area of ellipse [21].

TABLE 1. The related work of sleep classification algorithm.

Algorithm	Database	Number of Classes	Number of Features	Accuracy
SVM [12]	MIT-BIH	2	12	79.31%
			10	79.99%
Random Forest [13]	SHRSV	3	41	88.67%
Multi-Class Bayesian Network [14]	PSG	3	80	80%
		4		69%
Combination of ELM with PSO [15]	MIT-BIH	2	18	82.1%
		3		76.77%
		4		71.52%
		6		62.66%
Morphological [16]	MIT-BIH	3	15	77.02%
DNN [17]	MIT-BIH	3	11	77%

In 2013, Zong *et al.* [18] deployed WELM algorithm using weight matrix to solve the imbalanced class distribution. The weight matrix (W), which is a diagonal matrix with its elements, is shown in (1).

$$W = {}^{1}/_{\#(t_i)} \tag{1}$$

The use of the weight element helps the model to deal with the imbalanced class distribution by weakening the majority class and strengthening the minority class, also immediately compensating with the C-constant of the ELM optimization problem. The WELM optimization using the additional weight matrix is shown in (2).

$$\begin{aligned} &\text{minimize: } \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{1}{2}C\mathbf{W}\sum_{i=1}^{N}\|\xi\|^2 \\ &\text{subject to: } \mathbf{H}\boldsymbol{\beta} = t_i - \xi_i, \quad i = 1, \ldots, N \end{aligned} \tag{2}$$

Each class was mapped into an array with m elements for m multiclass classification. Each element had a value of either −1 or 1, where the targeted class had a value of 1. The WELM algorithm used for testing and training is illustrated in Fig. 3.

III. THE PROPOSED SYSTEM ARCHITECTURE AND SLEEP STAGE CLASSIFICATION ALGORITHM

This study aimed to develop a sleep monitoring system using event-driven and microservices architecture with satisfactory performance based on memory allocation, response time, and throughput. In addition, the combination of WELM and

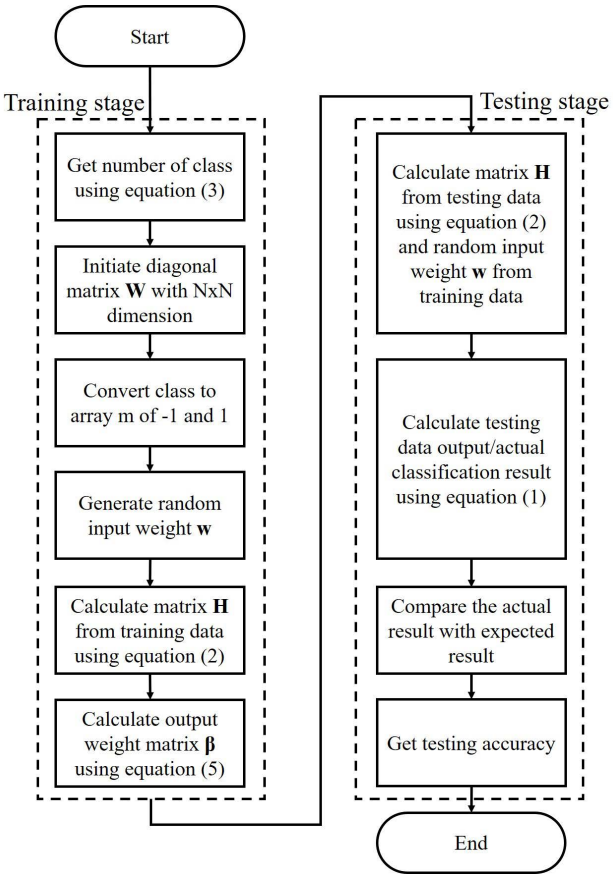


FIGURE 3. The WELM algorithm for testing and training.

PSO algorithms to achieve better sleep classification accuracy is also provided.

The sleep monitoring system via the IoT architecture with microservices and event-driven is shown in Fig. 4. The system consists of data acquisition and monitoring applications integrated into the IoT cloud server. The microservice applications on each IoT platform block had their source code, independent from other applications, and could be deployed separately.

The MIT-BIH polysomnographic data were streamed and used as sensor simulation data in the data acquisition process. The proposed IoT platform has five functions for sleep monitoring: 1) receiving incoming data and information from the sensor, 2) saving data to the database, 3) classifying the sleep stages, 4) measuring sleep quality, and 5) visualizing the sleep data to the dashboard.

The sensor gateway was deployed to receive data, whereas the sensor data were persistent for storing data. Message brokers were used for the communication. After receiving the sensor data, the sensor gateway publishes an event and stores it using sensor data persister. Data ingestion and storage were performed separately and asynchronously through service separation.

After receiving and saving the sensor data in the database, sleep stage classification was carried out using the WELM and PSO algorithm combination. The process also involves

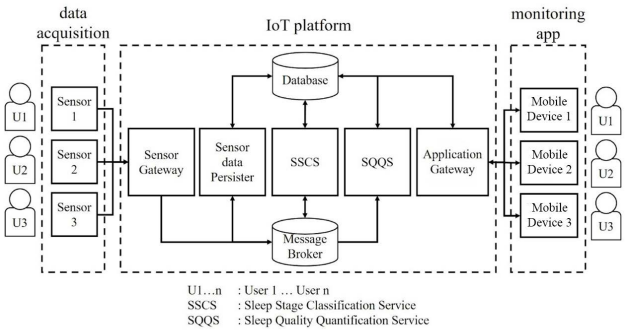


FIGURE 4. The sleep monitoring system via IoT architecture proposed in this study.

scheduling a system to check the available data in the database. An event was published when the classification process finished and responded to using a fuzzy algorithm to quantify sleep quality.

Finally, sleep quality data were be retrieved from the database and displayed on the dashboard. The proposed service also provides administrative features such as patient registration.

A. FIRST PART: THE SENSOR GATEWAY

An IoT platform related to sensors for sleep monitoring is a sensor gateway. The representational state transfer (REST) protocol [22] was used with an additional security layer using the JSON Web Token (JWT) [23], [24]. The patient was identified using the request body, which consisted of the ECG data and the patient's unique sensor token. The message broker was then received the data with "ecgDataInput" topic, as shown in Fig. 5. The asynchronous and separated process of the data storing and receiving process resulted from the event-driven and microservices architecture. The implementation expected a lower response time and a higher throughput in the sensor gateway.

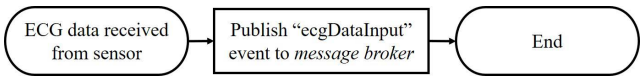


FIGURE 5. The flowchart of sensor gateway.

B. SECOND PART: THE SENSOR DATA PERSISTER

The "ecgDataInput" event was used to trigger the sensor data persister to validate and keep the data in the SQL database. The consumer group in the event-driven architecture was used as a concept for designing the sensor data persister; hence, replication to process large amounts of data from the sensor gateway was prepared.

Every patient had only one sleep quality data because only one set of ECG data was processed each day. The ECG data processing time was determined using a time range field during the patient registration process. Fig. 6 shows the flowchart of the sensor data persister.

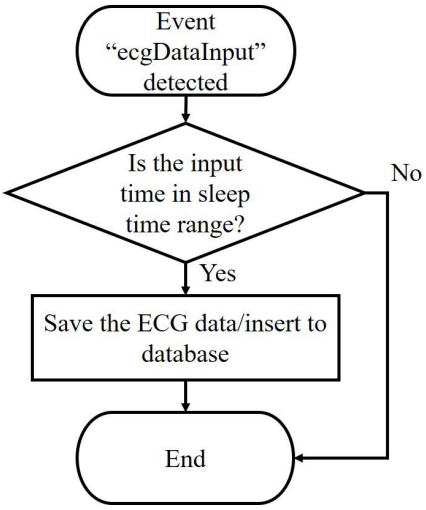


FIGURE 6. The flowchart of sensor data persister.

C. THIRD PART: SLEEP STAGE CLASSIFICATION SERVICE (SSCS)

The combined WELM and PSO algorithms were deployed for sleep stage classification as an upgrade to the combination of ELM and PSO algorithms by Lesmana *et al*. [15]. The ELM-PSO combination was deployed as a reference because of its quick learning time while maintaining a high generalization performance. The MIT-BIH database of human sleep ECG recordings [25] containing 10274 samples of 18 recording files was used as the dataset for classification. After removing invalid annotations or time, 10154 valid data were obtained. In every 30 seconds segment, six sleep stages were denoted in the MIT-BIH data samples, further mapped into three or four classes. The data distribution obtained for the three classes were wake (30.5%), REM (6.89%), and NREM (62.61%), whereas for the four classes were wake (30.5%), REM (6.89%), deep sleep (NREM3 and NREM4) (6.54%), and light sleep (NREM1 and NREM2) (56.07%). The dataset was then divided into 70:30 proportions as training and testing data, following the method described by Lesmana *et al.* [15].

Feature selection was carried out using the PSO algorithm to choose the best subset features because of the reduced selected features that affect processing time, resulting in a better and faster sleep stage classification [26]. The position of the PSO particle to represent the masked feature is designated by binary coding. The 18 features consist of seven frequency-domain features and 11 time-domain features [16]. The employed frequency-domain features are total power (TP), total power in the very low frequency range ($\leq$0.04 Hz) (VLF), low frequency range (0.04 Hz to 0.15 Hz) (LF), high frequency range (0.15 Hz to 0.4 Hz) (HF), ratio of LF and HF, LF, and HF power in normalized units. Meanwhile, the time-domain features are: 1) RR interval average (AVNN), 2) RR intervals' standard deviation (SDNN), 3) square root of the average squared differences of adjacent NN intervals (RMSSD), 4) standard deviation of differences between adjacent RR intervals (SDSD), 5) count of successive differences of RR intervals that are more than 50ms (NN50), 6) the division of NN50 and the total of RR intervals minus 100 (pNN50), 7) HRV triangular index, 8) standard deviation of points perpendicular to the axis of line-of-identity (SD1) and 9) along the axis of line-of-identity (SD2), 10) ratio of SD1 and SD2, and 11) area of ellipse [21]. Following the work of Malik *et al.* [21], 18 bits were deployed to represent 18 extracted features, yielding 262144 particles. For instance, a particle with a position of 215176 (110100100010001000 in binary) indicates that feature numbers 1, 2, 4, 7, 11, and 15 are the selected features.

Fig. 7 shows a flowchart of the combined WELM and PSO algorithms. In the testing dataset, model accuracy was the calculated fitness value. WELM was deployed to calculate the accuracy of each particle as a fitness function. A total of 80 sets of 20 particles with 20 iterations were used, whereas the c_1 and c_2 for PSO was 1.2 as stated in [15].

The flowchart can be explained as follows:

1. Dividing the dataset to 70:30 for the ratio of training and testing.
2. Initializing P particle with I iteration.
3. Initializing 262144 particle positions with the value of 0.
4. Decoding the position of the particle, obtaining, and masking each particle's features.
5. Calculating fitness value using WELM, which is used for testing accuracy.
6. Updating *pBest* (local) and *gBest*(global)
7. Updating particle velocity, including its new position.

Fig. 8 shows a flowchart of the SSCS scheduler and validation algorithm, in which two validations were run every second on the database. After the sleep time range, ECG data were no longer recorded for the day. Then, the classification process is carried out based on the testing phase in Fig. 3, with the w matrix, β matrix, and selected features of the learning process using a combination of WELM and PSO. The "*sleepStageReady*" event would be published by sleep stage classification service after finishing the sleep quality classification.

D. FOURTH PART: SLEEP QUALITY QUANTIFICATION SERVICE (SQQS)

The sleep quality level was quantified using a fuzzy algorithm[11] based on the awake, deep sleep, and total sleep duration percentage in the sleep quality quantification service (SQQS). Fig. 9 shows a flowchart of the event-based concept in the SQQS.

The process of sleep quality quantification was executed by the "*sleepStageReady*" event from the SSCS. The work of Ang *et al.* [11], which used the jFuzzyLogic library, was deployed in which the awake, deep sleep, and total sleep duration from the SSCS were used as fuzzy inputs. A set of data for metrics or extra information on the dashboard was calculated from the sleep stages in SQQS, such as the duration of awake, light sleep, deep sleep, total sleep, REM, as well as

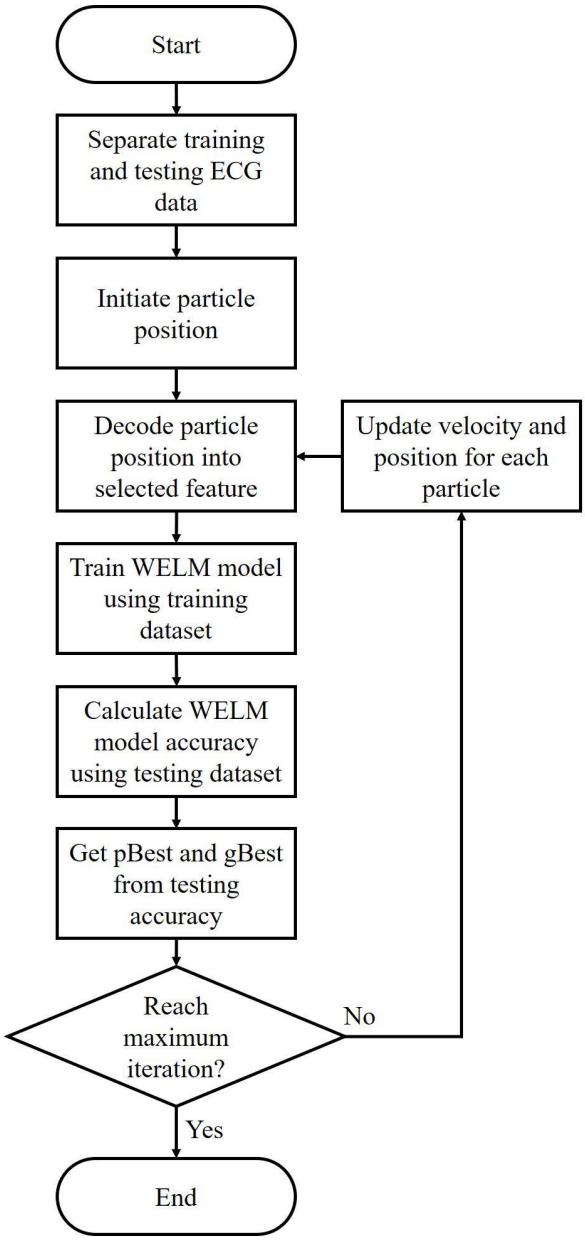


FIGURE 7. The flowchart of combined WELM and PSO algorithm.

wake time, time to sleep, and sleep efficiency, which are the percentages of total sleep duration. The completed result was then stored in the database.

E. FIFTH PART: APPLICATION GATEWAY

In addition to acting as a gateway for the application, it also facilitated the registration of patients and health practitioners. A sensor token is generated using a universally unique identifier (UUID) in the patient registration process [27]. REST endpoints were provided by this service to retrieve the ECG, sleep quality, and sleep stage information for each patient.

F. SIXTH PART: APPLICATION MONITORING

A mobile-based application for clients or patients is used for monitoring application. Tables and graphs were created to present informative sleep data on the dashboard regarding the details of patient sleep.

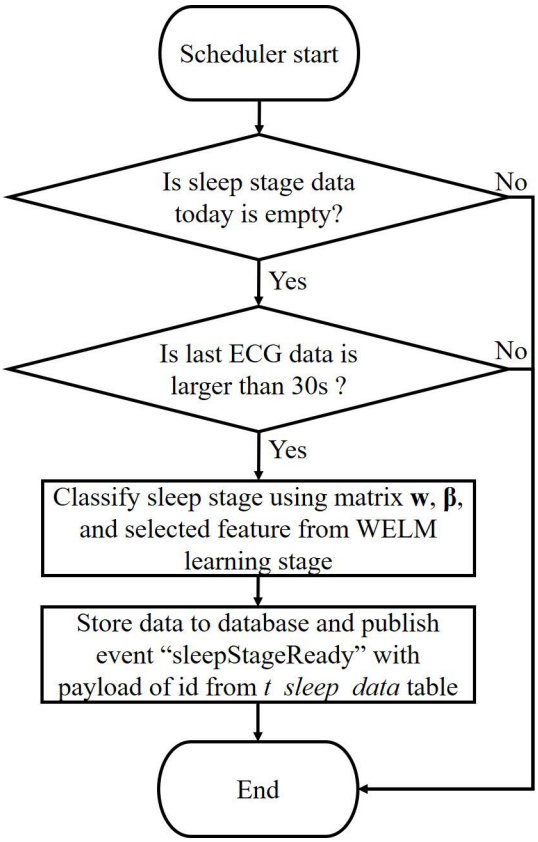


FIGURE 8. The flowchart of SSCS scheduler and validation algorithm.

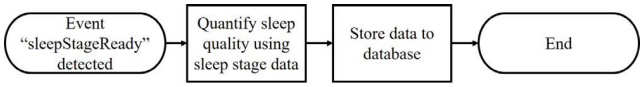


FIGURE 9. The flowchart of event-based concept in SQQS.

IV. PROPOSED MONITORING DASHBOARD

The proposed dashboard was built using the Flutter framework and had two dashboard types: the daily sleep stage dashboard and the overall sleep stage dashboard, as presented in Fig. 10 and 11. The dashboard has two types of user access: patient and health practitioner.

Daily sleep quality and sleep stage dashboards show sleep quality, sleep stage percentage, and sleep metrics according to the applied date or time filter. The sleep stages were presented using a graph and several colors to identify each stage, and the sleep metrics consisted of awake duration, deep sleep, light sleep, REM, total sleep, wake time, time to sleep, sleep quality, and sleep efficiency.

The overall quality and sleep stage dashboards contained information about overall patient sleep quality displayed in bar charts with sleep quality as the y-axis and day as the x-axis. The display of all data depended on the provided start and end date filters. Sleep metrics showing the duration of awake, deep sleep, light sleep, REM, and total sleep, as well

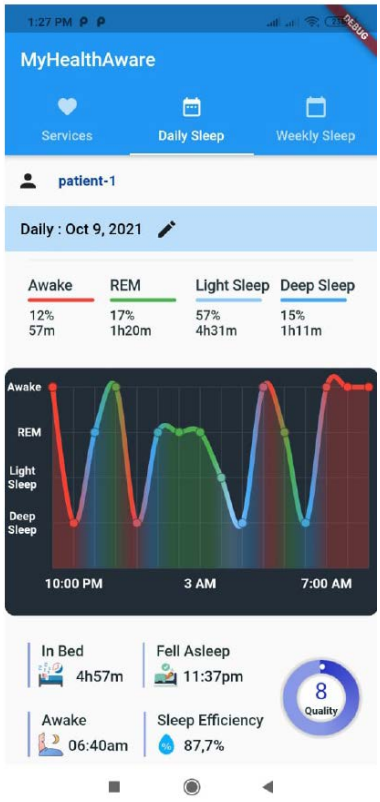


FIGURE 10. The dashboard of the daily sleep stages.

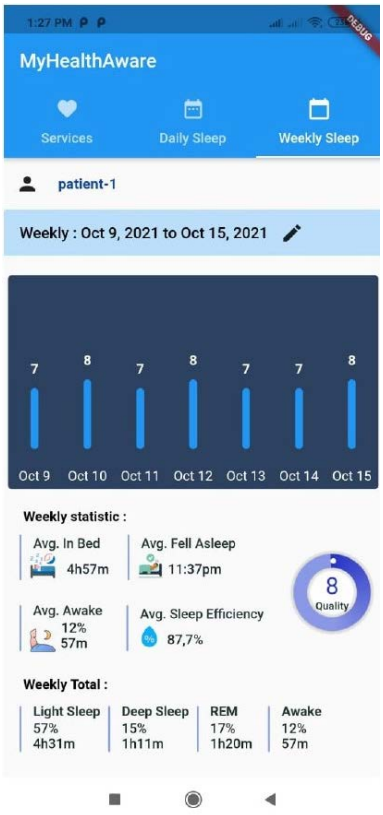


FIGURE 11. The dashboard of overall quality and sleep stages.

as time to sleep, wake time, sleep quality, and sleep efficiency were also shown.

V. RESULTS AND DISCUSSION

To evaluate the performance of the proposed IoT platform using microservices, event-driven, and the integration of WELM with PSO, four measurements were performed on a single machine with four CPUs equipped with 15 GB RAM and a clock rate of 2 GHz. The evaluations were performed as follows:

1) The first evaluation is related to the sensor gateway throughput and response time by measuring the number of transactions that the system can handle within a certain period. The number of transactions or requests per second was obtained for throughput evaluation. The duration of transaction or request being handled by a system was denoted as the response time, in which the measurement was started from the request sent to the server until the response was received [28]. A simulator streamed from the MIT-BIH database, was deployed to simulate the sensor data. A comparison to monolith and microservices architecture was made without event-driven or directly to the database with one and two replica applications.
2) RAM or memory usage evaluation of the proposed architectures was conducted and compared to monolith and microservices architecture with no event-driven architecture, or directly to the database using seven replica applications.
3) Execution time was evaluated for both the SQQS and SSCS. The speed of data processing from the sensor and the calculation of sleep quality data were determined.
4) Finally, the combined WELM and PSO algorithms for the imbalanced dataset were evaluated and compared to the previous method using the following metrics: model training and testing accuracy, sensitivity, specificity, and AUC. For the classification data, the annotations from the MIT-BIH database were used as a comparison.

A. THE EVALUATION OF THROUGHPUT AND RESPONSE TIME

Fig. 12 shows a comparison of the evaluated architectures. For testing purposes, 20000 requests were made using the Apache Jmeter [29]. The microservices and event-driven (McsED) architecture were compared to microservices without an event-driven architecture or directly to the database (McsDB) [29], as well as monolith architecture (MnlDB) [4], [5] with one and two service replicas. In summary, three architectures with two types of service replicas yielded six types of architecture.

Table 2, Fig. 13, and Fig. 14 present comparisons of throughput and response times. The proposed system

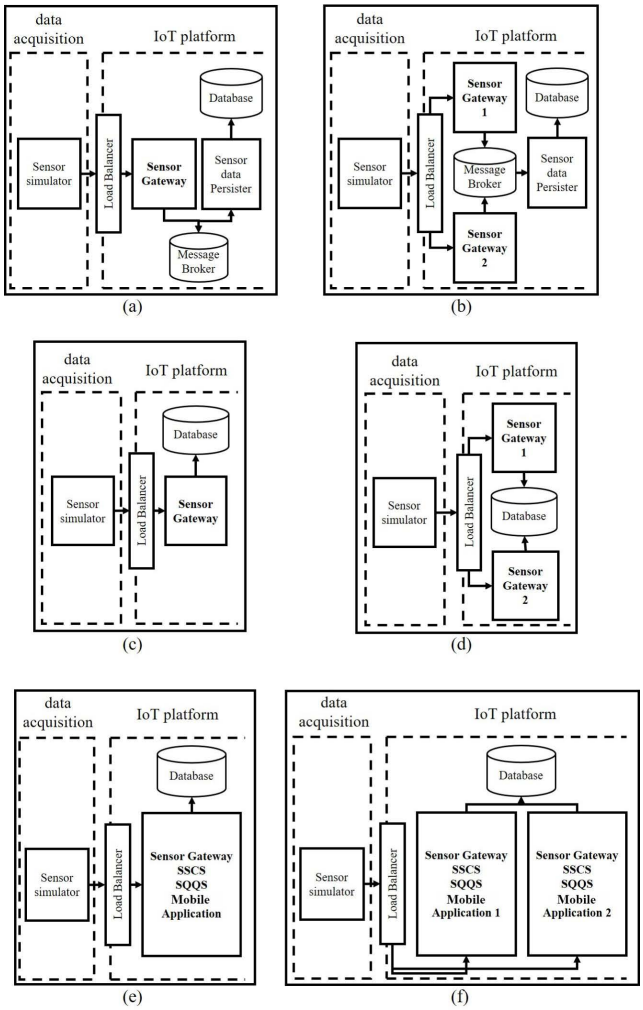


FIGURE 12. The comparison of the evaluated architectures. (a) 1McsED (proposed), (b) 2McsED (proposed), (c) 1McsDB [3], (d) 2McsDB [3], (e) 1MnlDB [4], [5], (f) 2MnlDB [4], [5].

TABLE 2. The comparison of throughput and response time.

Architecture[a]	Throughput (transaction per second)	Response time (ms)
1 McsED (proposed)	803	253.34
2 McsED (proposed)	1508	68.8
1 McsDB [3]	695	374.72
2 McsDB [3]	1119	155.84
1 MnlDB [4], [5]	476	596.22
2 MnlDB [4], [5]	783	280.6

[a]McsED = microservices and event-driven architecture, McsDB = microservices without event-driven architecture, MnlDB = monolith architecture connected directly to the database.

exhibited the best performance for one and two replicas. Using one replica, the proposed architecture could process 803 sensor data with a response time of 253.34 ms. On two replicas, 1508 transactions were achieved every second with a response time of 68.8 ms.

Compared to the monolith architecture, which was directly connected to the database (MnlDB), the throughput was

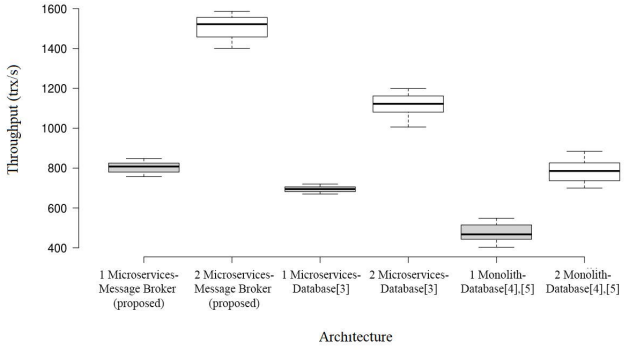


FIGURE 13. The evaluation of throughput using several architectures.

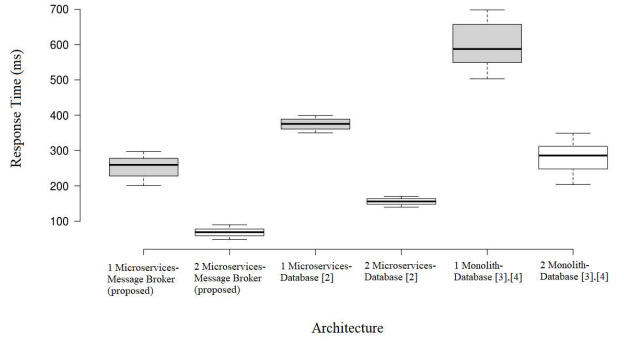


FIGURE 14. The evaluation of response time using several architectures.

increased by 92.59%, and the response time was decreased by 75.48% for two replicas owing to changes in microservices and event-driven architecture (McsED). The message broker implementation also increased the throughput by 34.76% and decreased the response time by 55.85% for two replicas compared to microservices without an event-driven architecture (McsDB). The proposed architecture improved the throughput and response time because the gateway sensor services were separated from other functions. Hence, transactions in the database were not required and they became more focused on performing tasks. Asynchronous data storage processing is possible using message broker, which is handled by the sensor data persister service. A lower performance was exhibited in the microservices without event-driven architecture (McsDB) because the sensor gateway required receiving and storing data in the database. The worst performance was shown in the monolith architecture (MnlDB) because of its requirement to provide all sleep quality monitoring functions in one large application.

B. THE COMPARISON OF MEMORY USAGE

Fig. 15 shows RAM or resource memory usage per architecture for up to 7 replicas of sensor-gateway replication. Equation (3) was used to calculate v, which increases the rate

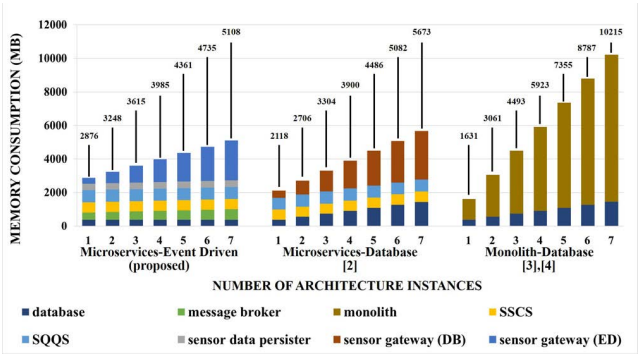


FIGURE 15. The architecture in average memory consumption per instance replication comparison.

in memory consumption per instance replication, where:

$$v = \frac{1}{N} \sum_{i=i}^{N} x_{i+1} - x_i \tag{3}$$

The most efficient memory usage was observed in the monolith architecture (MnlDB). In contrast, the highest memory consumption (2898 MB) was achieved in the proposed architecture (McsED).

The lowest memory usage for 2 to 4 replicas was achieved using McsDB. When the replica was further increased, the lowest memory consumption was obtained by the proposed architecture at 372 MB on average per added replica (v), compared to the microservices-database (McsDB) architecture and monolith architecture (MnlDB) at 592.5 MB and 1430.67 MB, respectively. Table 3 compares the rates of increase in memory usage per instance replication for the seven replicas.

TABLE 3. The rate of increase in memory usage per instances replication for 7 replicas comparison ($N = 7$).

Architecture	v (MB/replica)
McsED (proposed)	372
McsDB [3]	592.5
MnlDB [4], [5]	1,430.67

The microservices architecture provided low memory usage in the proposed method, in which the replication of the required services was enabled, namely, the sensor gateway. Sensor gateway replication cannot be performed separately in a single-entity application, such as a monolith architecture (MnlDB). Further reduction was made possible using event-driven architecture, which results in reducing the additional RAM resource usage for connecting to the database. The addition of a gateway sensor replica (DB) in the McsDB architecture increases the database memory usage owing to the direct connection of the gateway sensor to the database, thus requiring more allocated memory.

C. THE EVALUATION OF SSCS AND SQQS EXECUTION TIME

After a patient's ECG data were fully collected for one day, the execution times of the SSCS and SQQS were evaluated.

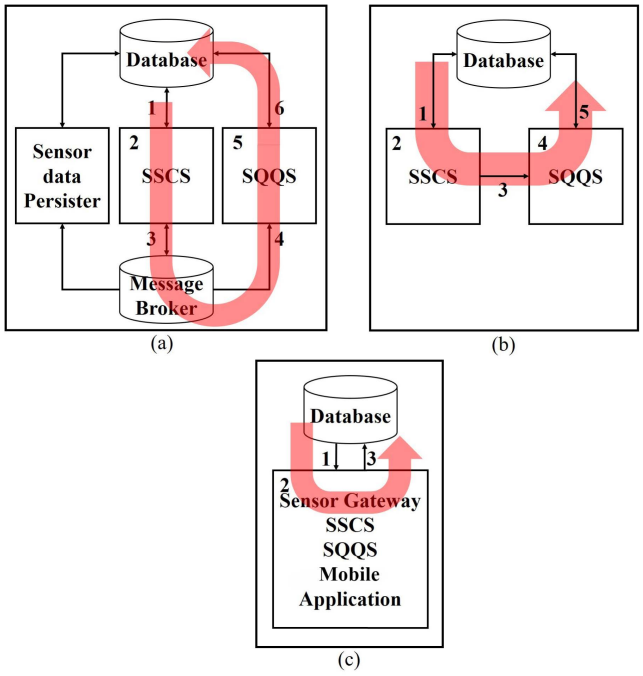


FIGURE 16. The comparison of data flow of ECG data for sleep quality processing in a) McsED, b) McsDB, c) MnlDB.

Fig. 16 shows the data flow of the ECG data for sleep quality processing. The slp41 file, which has 23400 ECG data (6 h 30 min long), was used as the ECG sample data, and the evaluation was performed 50 times.

The proposed architecture (McsED) consists of six steps to convert ECG data into sleep quality data. The first step was database reading using SSCS scheduler. Second, the ECG data in the sleep stage were classified using the SSCS. Third, the "sleepStageReady" event was published to the message broker by the SSCS. Fourth, the SSQS detected the "sleepStageReady" event. Fifth, the sleep stage was converted into sleep quality using the SQQS. Finally, the sleep quality data were stored in the database.

The processes of the microservices database architecture (McsDB) and proposed architecture (McsED) were similar except in the third and fourth steps, which were converted into one step. This step was the SSCS call to the SQQS API, triggering sleep quality quantification instead of sending an event.

In contrast to McsDB and McsED, the monolith-database architectures (MnlDB) only contained three steps. First, the SSCS scheduler detected that the ECG data were stored in the database. Second, SSCS and SQQS performed the sleep classification process and sleep quality quantification. Third, the sleep quality data were stored in the database.

Table 4 summarizes the average execution time comparison between WELM-PSO and ELM-PSO for each MIT-BIH record. The average execution time for the proposed architecture (McsED) was 5455 ms. In other words, the sleep quality data would be accessible to patients within ±5.5 ms after ECG data was collected. In comparison, the

TABLE 4. The comparison of average execution time between WELM-PSO AND ELM-PSO for each MIT-BIH record.

Architecture	Average execution time (ms)
McsED (proposed)	5455 ms
McsDB [3]	5028.7 ms
MnlDB [4], [5]	4511.32 ms

average execution time of microservices database (McsDB) and monolith database architecture (MnlDB) was 5028.7 ms and 4511.32 ms. The fastest processing time was achieved by using MnlDB as compared to other architectures because the processing in MnlDB was done in 1 source code. Hence, a remote function of the call procedure for other services is not necessary. Compared to the monolith and microservices-database architecture, the proposed architecture was slower by 17.3% and 7.81%, respectively, because the amount of time to send and receive events between services was small.

D. PERFORMANCE OF PROPOSED WELM AND PSO ALGORITHM

The combined WELM and PSO algorithm was tested for 3 classes such as awake, NREM, REM, and 4 classes like awake, light sleep, deep sleep, and REM classification.

By extracting ECG data, there were 18 features obtained from the MIT-BIH polysomnographic database. The 'strongest' features that affected the result were determined using the PSO algorithm, while testing accuracy for fitness values was using the WELM algorithm. More information about feature selection and assessment can be found in previous research published as conference proceedings [19].

The accuracy results using WELM and PSO for the three classes was 78.78% and four classes was 73.09%, respectively. Meanwhile, 10 features highly influenced the accuracy results are 1) AVNN, 2) RMSSD, 3) SDNN, 4) pNN50, 5) SDSD, 6) HRV Triangular Index, 7) SD1, 8) area of ellipse, 9) TP, 10) LF-HF ratio. The results of the testing accuracy for each subject file in the MIT-BIH polysomnographic database are shown in Table 5. WELM as a substitute for ELM in the work of Lesmana *et al.* [15] increased the testing accuracy by 1.76% for three classes and 1.57% for four classes.

The results proved that the theory by Zong *et al.* [18] provided different C values by using matrix W, which gave a larger and smaller C value for the minority and majority class, respectively. The reason for the utilization of matrix W was to provide single C values for the whole equation, either large or small, while poor performance or poor generalization still occurred in the minority classes[18]. Tables 6 and 7 provide performance comparisons for three and four classes of related works and the combined WELM and PSO algorithms in this study. Compared to previous works [15]–[17], the combination of WELM and PSO in this study resulted in better performance in terms of accuracy and number of features for three and four classes.

TABLE 5. The performance comparison between WELM-PSO and ELM-PSO [15] for each MIT-BIH record.

Record	3 class Testing Acc (%)		4 class Testing Acc (%)	
	WELM-PSO	ELM-PSO [15]	WELM-PSO	ELM-PSO [15]
slp01a	95.71	97.14	75.36	81.16
slp01b	71.02	68.22	67.42	68.22
slp02a	82.25	80.95	77.11	77.14
slp02b	77.94	79.75	76.08	78.48
slp03	74.68	73.21	65.14	62.20
slp04	76.63	76.17	74.89	72.30
slp14	66.09	64.15	66.15	63.21
slp16	67.11	61.17	65.18	58.25
slp32	86.45	84.82	75.04	75.92
slp37	91.38	90.87	89.42	90.87
slp41	67.8	60.09	67.3	57.76
slp45	84.11	75.56	67.2	62.95
slp48	84.51	83.63	82.66	84.00
slp59	78.45	73.53	67.08	53.68
slp60	83.25	77.51	79.9	77.51
slp61	76.89	75.81	68.44	63.08
slp66	85.5	77.10	78.15	78.46
slp67x	86.66	82.22	86.66	82.22

TABLE 6. The performance comparison between related works and WELM-PSO for 3 classes.

Authors	Number of features	Accuracy
Lesmana [15]	18	76.77%
Wei [17]	11	77%
Yücelbaş [16]	15	77.02%
Our work	**10**	**78.78%**

TABLE 7. The performance comparison between related works and WELM-PSO FOR 4 classes.

Authors	Number of features	Accuracy
Lesmana [15]	18	71.52%
Our work	**10**	**73.09%**

TABLE 8. Highest results of testing sensitivity, specificity, and AUC for 3 classes and 4 classes.

Metrics	3 class	4 class
Testing Sensitivity (%)	55	40
Testing Specificity (%)	84	87
Testing Area Under ROC Curve (AUC)	0.89	0.89

To enhance the evaluation of the WELM-PSO combination, the sensitivity, specificity, AUC of the model from the ROC curves were also calculated. The highest results of the 18 samples for three classes and four classes are shown in Table 8, while the ROC curves for three and four classes are shown in Fig. 17 and 18.

The AUC ranges from 0.5 to 1. According to Khouli et al. [30], a classification model can be considered excellent if the AUC value ranges from 0.9 to 1. Our study

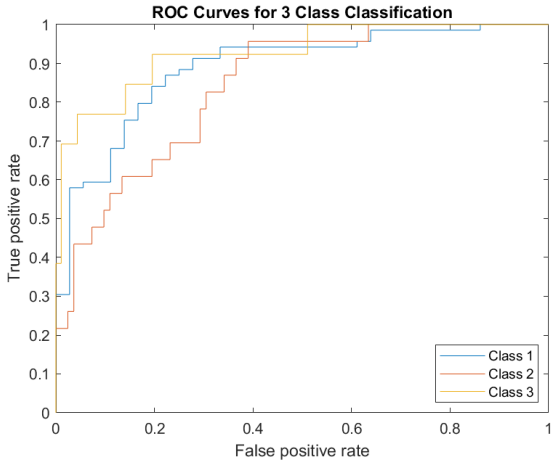


FIGURE 17. ROC curves of the highest AUC result for 3 classes.

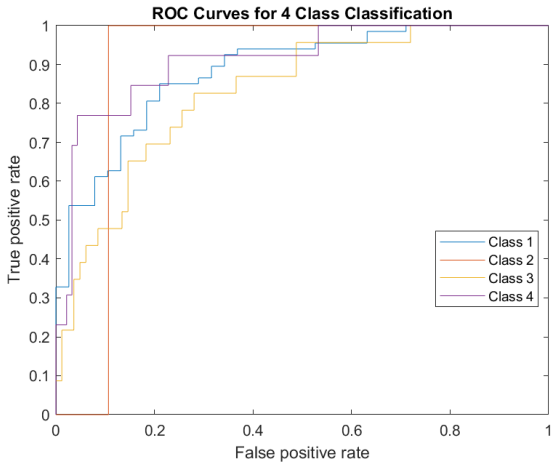


FIGURE 18. ROC curves of the highest AUC result for 4 classes.

has the highest AUC values of 0.89 for both the three and four classes classification, which can be considered acceptable.

VI. CONCLUSION

This paper presents a study on IoT architecture for sleep quality monitoring and a combination of WELM with PSO for sleep stage classification. The entire system consists of two main concepts: event-driven and micro-services in the IoT platform, which then splits again into five micro-services: 1) sensor gateway, 2) sensor data persister, 3) sleep stage classification, 4) sleep quality quantification, and 5) application gateway. The communication between these services was triggered by events in an asynchronous manner between the message broker and services.

The processing time of 803 ECG requests per second was achieved using a single replica of the proposed system with a response time of 253.34 ms, whereas 1508 requests per second were also obtained using two replicas with a response time of 68.8 ms. Initially, the proposed system consumed the highest resources. As the number of replicas increased, the memory consumption has the lowest increase when compared to other systems since microservices have independently separated services. The system used 372 MB per replica for the sensor gateway replication in the average memory consumption per instance replication. Testing accuracies of 78.78% and 73.09% were obtained from the WELM-PSO algorithm, which were higher than other studies for three and four classes. In addition, the AUC of the proposed model was also calculated, and the highest result reached both 0.89 for the three and four class classification.

Despite these advantages, the data processing from raw ECG data into sleep quality data using the microservices and event-driven architecture was 7.81% and 17.3% slower than the other architectures, respectively, owing to the event-driven communication system. However, the data flow in the proposed architecture can be improved by adding sensor transmission protocol and event compression. For future work, additional features obtained from other sensors may improve the accuracy of sleep stage classification, such as respiration sensors or actimetry. Moreover, the suggestion from the reviewer about using the sleep microstructure over the sleep macrostructure to assess sleep quality is also considered for further research.

REFERENCES

[1] W. H. Moorcroft, *Understanding Sleep and Dreaming*, 2nd ed. New York, NY, USA: Springer, 2013.

[2] 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," *Proc. Comput. Sci.*, vol. 157, pp. 321–328, Oct. 2019, doi: 10.1016/j.procs.2019.08.173.

[3] M. Kim, M. Asthana, S. Bhargava, K. K. Iyyer, R. Tangadpalliwar, and J. Gao, "Developing an on-demand cloud-based sensing-as-a-service system for Internet of Things," *J. Comput. Netw. Commun.*, vol. 2016, pp. 1–17, Aug. 2016, doi: 10.1155/2016/3292783.

[4] K. N. Swaroop, K. Chandu, R. Gorrepotu, and S. Deb, "A health monitoring system for vital signs using IoT," *Internet Things*, vol. 5, no. 1, pp. 116–129, 2019, doi: 10.1016/J.IoT.2019.01.004.

[5] S. Mohapatra, S. Mohanty, and S. Mohanty, "Smart healthcare: An approach for ubiquitous healthcare management using IoT," in *Big Data Analytics for Intelligent Healthcare Management*. New York, NY, USA: Academic, 2019, ch. 7, pp. 175–196.

[6] J. Thönes, "Microservices," *IEEE Softw.*, vol. 32, no. 1, p. 116, Jan. 2015, doi: 10.1109/MS.2015.11.

[7] B. M. Michelson, *Event-Driven Architecture Overview*. Boston, MA, USA: Patricia Seybold Group, Feb. 2006. [Online]. Available: https://www.omg.org/soa/Uploaded%20Docs/EDA/bda2-2-06cc.pdf

[8] A. Mouttham, L. Peyton, B. Eze, and A. E. Saddik, "Event-driven data integration for personal health monitoring," *J. Emerg. Technol. Web Intell.*, vol. 1, no. 2, pp. 110–118, Nov. 2009, doi: 10.4304/jetwi.1.2.110-118.

[9] P. Niblett and S. Graham, "Events and service-oriented architecture: The OASIS web services notification specification," *IBM Syst. J.*, vol. 44, no. 4, pp. 869–886, 2005, doi: 10.1147/sj.444.0869.

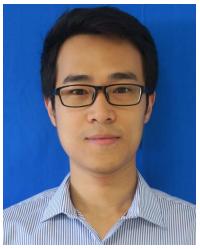
[10] D. J. Buysse, C. F. Reynolds, T. H. Monk, S. R. Berman, and D. J. Kupfer, "The Pittsburgh sleep quality index: A new instrument for psychiatric practice and research," *Psychiatry Res.*, vol. 28, pp. 193–213, May 1989, doi: 10.1016/0165-1781(89)90047-4.

[11] C. K. Ang, W. Y. Tey, P. L. Kiew, and M. Fauzi, "An artificial intelligent approach using fuzzy logic for sleep quality measurement," *J. Mech. Eng.*, vol. 4, pp. 31–47, Sep. 2017.

[12] M. Adnane, Z. Jiang, and Z. Yan, "Sleep–wake stages classification and sleep efficiency estimation using single-lead electrocardiogram," *Expert Syst. Appl.*, vol. 39, no. 1, pp. 1401–1413, Jan. 2012, doi: 10.1016/j.eswa.2011.08.022.

[13] M. Xiao, H. Yan, J. Song, Y. Yang, and X. Yang, "Sleep stages classification based on heart rate variability and random forest," *Biomed. Signal Process. Control*, vol. 8, no. 6, pp. 624–633, 2013, doi: 10.1016/j.bspc.2013.06.001.
[14] P. Fonseca, X. Long, M. Radha, R. Haakma, R. M. Aarts, and J. Rolink, "Sleep stage classification with ECG and respiratory effort," *Physiolog. Meas.*, vol. 36, no. 10, pp. 2027–2040, Oct. 2015, doi: 10.1088/0967-3334/36/10/2027.
[15] T. F. Lesmana, S. M. Isa, and N. Surantha, "Sleep stage identification using the combination of ELM and PSO based on ECG signal and HRV," in *Proc. 3rd Int. Conf. Comput. Commun. Syst. (ICCCS)*, Apr. 2018, pp. 258–262, doi: 10.1109/CCOMS.2018.8463307.
[16] Ş. Yücelbaş, C. Yücelbaş, G. Tezal, S. Özşen, and Ş. Yosunkaya, "Automatic sleep staging based on SVD, VMD, HHT and morphological features of single-lead ECG signal," *Expert Syst. Appl.*, vol. 102, pp. 193–206, Jul. 2018, doi: 10.1016/j.eswa.2018.02.034.
[17] R. Wei, X. Zhang, J. Wang, and X. Dang, "The research of sleep staging based on single-lead electrocardiogram and deep neural network," *Biomed. Eng. Lett.*, vol. 8, no. 1, pp. 87–93, Feb. 2018, doi: 10.1007/s13534-017-0044-1.
[18] W. Zong, G.-B. Huang, and L. Chen, "Weighted extreme learning machine for imbalance learning," *Neurocomputing*, vol. 101, pp. 229–242, Feb. 2013, doi: 10.1016/j.neucom.2012.08.010.
[19] N. Surantha, O. K. Utomo, and S. M. Isa, "High-performance and resource-efficient IoT-based sleep monitoring system," in *Proc. IEEE 91st Veh. Technol. Conf. (VTC-Spring)*, May 2020, pp. 1–5, doi: 10.1109/VTC2020-Spring48590.2020.9129521.
[20] P. Cingolani and J. Alcalá-Fdez, "jFuzzyLogic: A Java library to design fuzzy logic controllers according to the standard for fuzzy control programming," *Int. J. Comput. Intell. Syst.*, vol. 6, pp. 61–75, Jun. 2013, doi: 10.1080/18756891.2013.818190.
[21] M. Malik, J. T. Bigger, A. J. Camm, R. E. Kleiger, A. Malliani, A. J. Moss, and P. J. Schwartz, "Heart rate variability. Standards of measurement, physiological interpretation, and clinical use," *Eur. Heart J.*, vol. 17, no. 3, p. 381, Mar. 1996, doi: 10.1093/oxfordjournals.eurheartj.a014868.
[22] C. Pautasso, "RESTful web services: Principles, patterns, emerging technologies," in *Web Services Foundations*. New York, NY, USA: Springer, 2013, ch. 2, pp. 31–51.
[23] Y. Yang, Q. Zu, P. Liu, D. Ouyang, and X. Li, "MicroShare: Privacy-preserved medical resource sharing through MicroService architecture," *Int. J. Biol. Sci.*, vol. 14, no. 8, pp. 907–919, 2018, doi: 10.7150/ijbs.24617.
[24] X. Larrucea, I. Santamaria, R. Colomo-Palacios, and C. Ebert, "Microservices," *IEEE Softw.*, vol. 35, no. 3, pp. 96–100, May 2018, doi: 10.1109/ms.2018.2141030.
[25] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, Jun. 2000, doi: 10.1161/01.cir.101.23.e215.
[26] B. Xue, M. Zhang, W. N. Browne, and X. Yao, "A survey on evolutionary computation approaches to feature selection," *IEEE Trans. Evol. Comput.*, vol. 20, no. 4, pp. 606–626, Aug. 2016, doi: 10.1109/tevc.2015.2504420.
[27] P. Leach, M. Mealling, and R. Salz. (2005). *A Universally Unique IDentifier (UUID) URN Namespace*. [Online]. Available: http://www.ietf.org/rfc/rfc4122.txt
[28] M. M. Arif, W. Shang, and E. Shihab, "Empirical study on the discrepancy between performance testing results from virtual and physical environments," *Empirical Softw. Eng.*, vol. 23, no. 3, pp. 1490–1518, Oct. 2017, doi: 10.1007/s10664-017-9553-x.
[29] A. S. Foundation. (2013). *Apache JMeter Glossary*. [Online]. Available: http://jmeter.apache.org/usermanual/glossary.html
[30] R. H. El Khouli, K. J. Macura, P. B. Barker, M. R. Habba, M. A. Jacobs, and D. A. Bluemke, "Relationship of temporal resolution to diagnostic performance for dynamic contrast enhanced MRI of the breast," *J. Magn. Reson. Imag.*, vol. 30, no. 5, pp. 999–1004, Nov. 2009, doi: 10.1002/jmri.21947.

NICO SURANTHA (Member, IEEE) received the B.Eng. and M. Eng. degrees from the Institut Teknologi Bandung, Indonesia, in 2007 and 2009, respectively, and the Ph.D. degree from the Kyushu Institute of Technology, Japan, in 2013. He currently works as a full-time Lecturer with the Department of Electrical, Electronic and Communication Engineering, Tokyo City University, Japan. He is also a part-time Lecturer with the Computer Science Department, BINUS Graduate Program, Bina Nusantara University, Indonesia. His research interests include ubiquitous computing, intelligent systems, the Internet of Things, health monitoring, and system on chip design.



OEI K. UTOMO received the bachelor's degree from the Department of Electrical Engineering, Satya Wacana Christian University, Salatiga, in 2016. He is currently pursuing the degree with the Computer Science Department, BINUS Graduate Program—Master of Computer Science. His research interests include health monitoring and the Internet of Things.

EARLICHA M. LIONEL is currently pursuing the bachelor's and master's degrees in information technology with BINUS University. Her research interests include virtual reality and machine learning.

ISABELLA D. GOZALI is currently pursuing the bachelor's and master's degrees in information technology with BINUS University. Her research interests include virtual reality and machine learning.

SANI M. ISA (Member, IEEE) received the bachelor's degree from Padjadjaran University, Bandung, Indonesia, and the master's and Ph.D. degrees in computer science from the University of Indonesia. He is currently an Associate Professor with the Computer Science Department, BINUS Graduate Program—Master of Computer Science. He has some experience in teaching and research in remote sensing and biomedical engineering areas.