Journal of Computer Science and Electrical Engineering
ISSN: 2663-1938
DOI: 10.61784/jcsee3013

CLOUD BASED SLEEP QUALITY MONITORING AND EVALUATION SYSTEM

Xu Chen, DeQuan Kong*
School of Information and Electronic Technology, Jiamusi University, Jiamusi 154007, Heilongjiang, China.
Corresponding Author: DeQuan Kong, Email: 893907285@qq.com

Abstract: To facilitate the monitoring and assessment of sleep conditions, this paper designs a cloud-based sleep quality monitoring and evaluation system. The system comprises a signal acquisition and processing module and a cloud platform. The signal acquisition and processing module connects to the cloud platform via Bluetooth, enabling the collection, reception, processing, and storage of various signals during sleep. The cloud platform employs a Naive Bayes sleep staging algorithm to classify sleep stages and assess sleep quality.
Keywords: Non-contact sleep monitoring; Naive bayes sleep staging algorithm; Smart device terminal; PVDF

1 INTRODUCTION

Sleep is a crucial natural physiological need. A survey in the "2022 China National Health Sleep White Paper" found that over half of the respondents have experienced sleep problems, which have gradually become a significant factor impacting people's work and daily life. This indicates that sleep issues are increasingly prevalent among residents in China. However, fewer than 6% of hospitals nationwide have established sleep monitoring facilities, and the quality of existing sleep medicine centers varies, failing to meet the public's demand for sleep quality services. Sleep staging is fundamental to sleep research and a key step in assessing sleep quality. Therefore, designing an affordable and user-friendly sleep quality monitoring and evaluation system is of great practical significance both nationally and globally at this stage[1]. This paper aims to develop an intelligent monitoring system capable of assessing sleep quality, utilizing non-intrusive sensor technology to monitor subjects during normal sleep[2]. The hardware of the system performs real-time acquisition of physiological signals, while the software handles data processing and analysis[3], ultimately generating a sleep quality evaluation report.

2 SYSTEM OVERALL PLAN

The overall architecture of the cloud-based sleep quality monitoring and evaluation system is shown in Figure 1.
The system's smart device terminal, heterogeneous network transmission, cloud server, and web client are each designated for specific functions: sleep assistance, monitoring of sleep physiological data, intelligent data analysis, anomaly detection and alerting, and customized intelligent features.

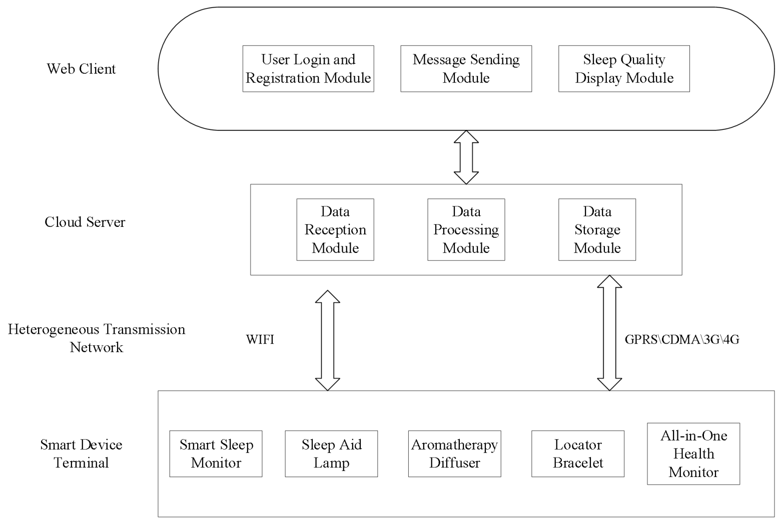

Figure 1 Overall Architecture of the Sleep Quality Detection System

To reduce software development complexity and make application development more flexible and efficient, a three-tier software architecture, as shown in Figure 2, has been designed. This architecture consists of a user interface layer, a business processing layer, and a data analysis layer. The distributed three-tier architecture aligns with the software development principle of "low coupling, high cohesion" [4]. This design separates interface display, business processing, and data support.

The user interface layer, which is the web page, allows users to interact directly with the application. The business processing layer is the core of the software; this project will use an improved Naive Bayesian sleep staging algorithm [5] to analyze physiological sleep data within a 1-minute window, determine the current sleep phase of the user, and generate a corresponding sleep quality report after a sleep monitoring session is completed. The data service layer is primarily responsible for collecting information from the smart device terminal, handling database operations, storage, and processing data requests from the business processing layer.

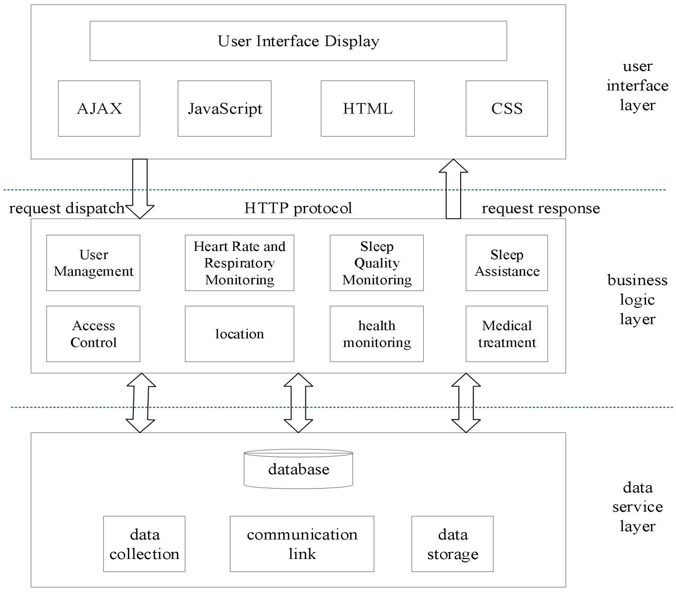

Figure 2 Sleep Monitoring Software Architecture

3 IMPROVEMENTS TO THE NAÏVE BAYES SLEEP STAGING ALGORITHM

In traditional Naive Bayes sleep staging computations, heart rate, respiratory rate, and heart rate variability are used [6]. Building upon this, the Naive Bayes classification method evaluates different types of data across four sleep stages and selects the most prevalent type as the final evaluation metric. However, there is a significant probability that the classification of the four stages may occur in pairs, presenting a dilemma for traditional Naive Bayes sleep staging methods, which may not provide intelligent classification.

To ensure the traditional Naive Bayes sleep staging algorithm functions correctly and to avoid issues such as those caused by high heart rate leading to respiratory rate problems, low heart rate or low respiratory intensity, and 0.5:0.5 sleep phase distribution due to high heart rate and respiratory intensity, improvements have been made. The improved Naive Bayes sleep staging algorithm process is shown in Figure 3.

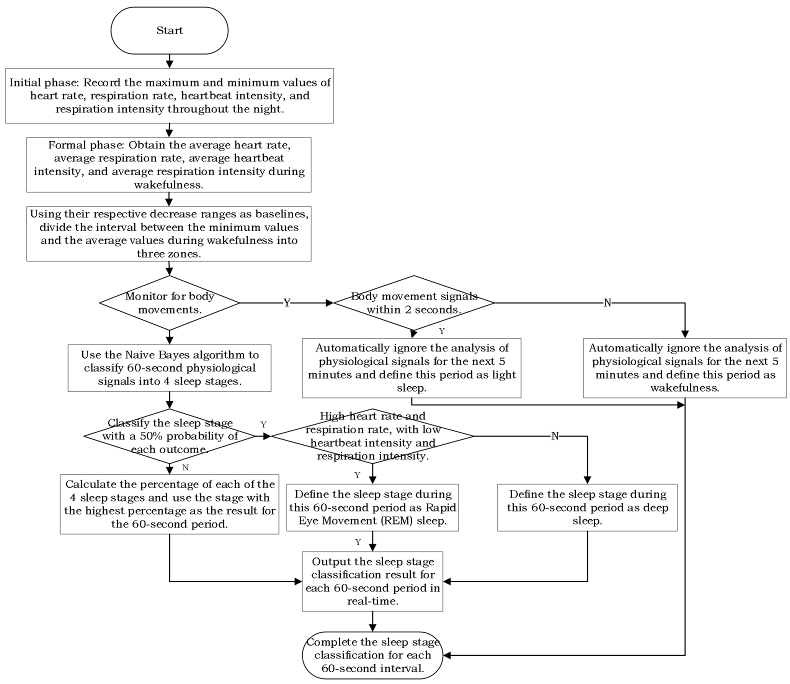

Figure 3 Naive Bayes Sleep Stage Classification Algorithm Flowchart

4 DESIGN OF THE CLOUD SERVER

The main functions of the cloud server are to receive, store, and process the sleep physiological information sent by the smart sleep monitor. The cloud server is primarily composed of three components: reception, storage, and processing. The data flow within the cloud server is illustrated in Figure 4.

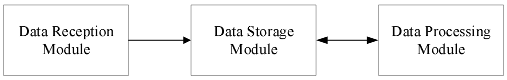

Figure 4 Data Flow to Cloud Server

4.1 Data Processing Logic of the Data Reception Module

After the Gateway Worker module of the cloud server receives real-time sleep physiological data sent by the smart sleep monitor [7], it processes the data primarily through three event callback functions in `Applications/Your App/Events.php`. When the smart sleep monitor establishes a Socket connection with the cloud server, the `on-Connect` event is triggered. The callback function for this event displays the login information of the smart device on the cloud server's control panel. When the smart sleep monitor disconnects from the cloud server, the `on-Close` event is triggered, and the `Close` function displays the device's logout information on the cloud server's control panel. Each time the smart sleep monitor sends physiological data to the cloud server, an `on-Message` event is triggered. The processing logic for this callback function is illustrated in Figure 5.

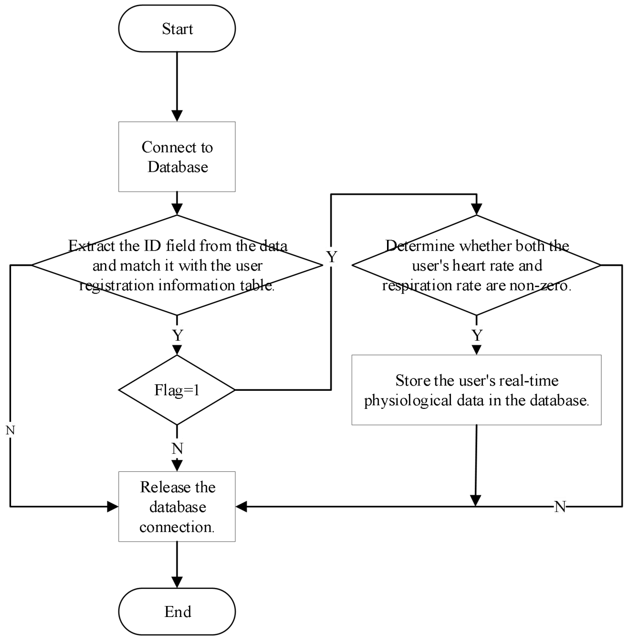

Figure 5 Data Processing Logic Diagram for the Data Reception Module

4.2 Design of the Data Storage Module

The data storage module is primarily used to store user information, such as basic user details and sleep quality assessment parameters.

4.3 Design of the Data Processing Module

Since the core module of the sleep monitoring software is the sleep monitoring module, the focus is on implementing the improved Naive Bayes sleep staging algorithm. This algorithm analyzes the user's physiological sleep data every minute to determine the current sleep phase and generates a sleep quality report at the end of each sleep monitoring session. The relevant data is then stored in a historical records table. Due to development challenges and proficiency issues, the backend data processing module of the system is implemented using PHP, with Sublime Text 3 as the development tool. The relationship between the sleep monitoring software and the data processing module is illustrated in Figure 6.

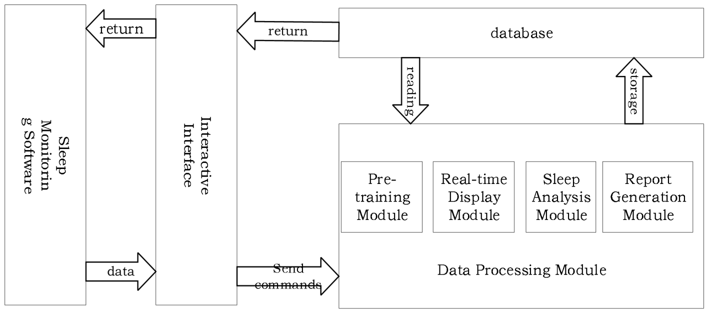

Figure 6 Relationship between Sleep Monitoring Software and Data Processing Module

(1) The functions of the preliminary training phase are mainly implemented by the preliminary training module. This involves collecting the maximum and minimum values of parameters such as heart rate, respiratory rate, and respiratory intensity throughout the entire sleep period of the subject. Measurements of these maximum and minimum values are taken, and the average heart rate, respiratory rate, heart intensity, and respiratory intensity for the first 5 minutes are assessed.
(2) The functions of the official sleep monitoring phase are accomplished through the real-time display module. This module acquires real-time physiological sleep data from the user, analyzes their sleep state, and tracks statistics on their movements, such as turning over, breathing pauses, getting out of bed, and waking times. Additionally, when determining the user's body movement status, the current sleep stage is identified based on the duration of body movements, without considering the physiological sleep signals from the subsequent 5 minutes. Based on this, the user's current state is assessed according to their actual conditions and the set alarm thresholds.
(3) The function of detecting body movement signals and the period beyond the 5 minutes after body movement cessation is mainly handled by the sleep analysis module. Using the Bayesian algorithm, the module determines the sleep phases within 1-minute segments of body movement signals, and performs statistical analysis to obtain the user's real-time sleep phase.
(4) The report generation module produces a complete sleep quality report after each sleep monitoring session. It also stores the relevant data from the sleep quality report in a historical records table, allowing users to easily query their past sleep data.

5 SYSTEM FUNCTION TESTING

Based on the actual application scenarios and the development progress of the sleep monitoring software, the testing process for this software has been designed to include three main phases: unit testing, integration testing, and system testing. The details are as shown in Table 1.

Table 1 Software Testing Process Design Table

Phase	Testing Method	Style
1	Unit Testing	Unit testing verifies whether the smallest units of software design—such as program modules or functional modules—contain errors, correctly implement their functions, and meet performance and interface requirements. Unit testing is typically performed using white-box testing techniques.
2	Integration Testing	Integrate the tested unit modules into a system or subsystem and then perform testing, focusing on the interfaces between different modules. Integration testing is typically conducted using black-box testing as the primary approach, supplemented by white-box testing.
3	System Testing	After completing unit and integration testing, conduct joint testing with other system components, such as data collection gateways and smart device terminals. The main goal of system testing is to verify the correctness of data interactions and response times between different components

Testing of various functional modules of the software is conducted on the established testing platform. During the functional testing, issues such as interface display and browser compatibility are also checked. For example, real-time monitoring is tested, as shown in Table 2.

Table2 Real-time Monitoring Test Case Table

Test Case Name	Real-time Monitoring
Test Objective	1.Verify that the system can display the user's current sleep physiological data in real-time.

	2.Verify whether the system can receive anomaly alerts based on user-defined alarm thresholds. 3.Verify whether the system can assess the overall quality of the user's sleep segment when the sleep monitoring session ends.
Test Steps	1.Start the Gateway Worker module on the cloud server to receive data sent by the smart sleep monitor. 2.Log in using the sleep test subject's account and navigate to the "Real-time Monitoring" page. 3.Click the "Start Monitoring" button and compare the real-time data displayed on the page with the raw data on the Gateway Worker control interface. Check if data exceeding the alarm thresholds is highlighted in red. 4.Click the "End Monitoring" button and verify if the sleep quality analysis section on the page generates the appropriate chart report.

When the user is ready to start sleep monitoring, the Gate-way Worker module on the cloud server is activated to receive data sent by the smart sleep monitor. At this point, the user logs into the Web client using their sleep monitoring account, navigates to the "Real-time Monitoring" page, and clicks the "Start Monitoring" button. The start time is recorded on the page, the status changes to "Monitoring," and the data on the page is updated in real-time to match the update speed of the raw data on the cloud server's Gate-way Worker control interface. Any real-time data exceeding the custom alarm threshold is highlighted in red. When the user's sleep session ends, clicking the "End Monitoring" button generates a corresponding chart report in the sleep quality analysis section of the page, which matches the data recorded in the database table `sm_sleep_report`. This functionality test is passed.

This project provides a comprehensive introduction to the overall architecture of the sleep monitoring system, presenting a detailed design for a non-contact sleep quality monitoring system based on a cloud platform. It also explains the design concepts of each functional component in relation to practical conditions. The accuracy and timeliness of monitoring sleep quality are directly related to the detection quality of heart rate and respiratory rate [8]. The design described in this paper meets the demand for an affordable and user-friendly sleep quality monitoring device, offering excellent application prospects and high market value.

COMPETING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

FUNDING

The project was supported by Sleep Quality Monitoring and Evaluation System Based on Cloud Platform (202210222139).

REFERENCES

[1] Zeng Jin, Lao Zhihui, Xu Mianqu. Design of an Embedded Sleep Posture Monitoring System Based on Infrared Cameras. Electronics Technology & Software Engineering, 2019(15): 168-170.
[2] Yang Jiafeng, Tong Jijun, Jiang Lurong, et al. Design and Implementation of a Non-Intrusive Sleep Respiratory Monitoring System. Computer Engineering and Applications, 2022, 58(05): 302-308.
[3] Liang Chao, Wang Peng, Cao Beibei, et al. Wearable Intelligent Sleep Quality Detection System. Journal of Electronic Measurement and Instrumentation, 2018, 32(05): 159-167.
[4] Yu D, Chander A, Islam N, et al. JavaScript instrumentation forbrowsersecurity. Acm Sigplan Notices, 2007, 42(1): 237-249.
[5] Jennifer Sloboda, Manohar Das. A Simple Sleep Stage Identification Technique for Incorporation in Inexpensive Electronic Sleep Screening Devices. Aerospace and Electronics Conference, Proceedings of the 2011 IEEE National, 2011: 21-24.
[6] Wu Taofeng, Gan Shuqi, Ouyang Chengjun, et al. Portable Sleep Quality Monitoring Device Based on Arduino. Electronics Technology & Software Engineering, 2021(18): 61-62.
[7] Zhong Linqian, Huang Jun, Liu Can, et al. Design and Implementation of Cloud-Based Sleep Monitoring Software. Information Communication, 2019, (03): 100-103.
[8] Dai Fengzhi, Lu Peng, Zhu Yuxuan. Design of a Multi-Sensor Sleep Monitoring and Assessment System. Foreign Electronic Measurement Technology, 2022, 41(04): 126-133.