Internet-of-Things-Enabled Data Fusion Method for Sleep Healthcare Applications

Fan Yang, Qilu Wu, Xiping Hu, Jiancong Ye, Yuting Yang, Haocong Rao, Rong Ma, and Bin Hu, *Senior Member, IEEE*

***Abstract*—The Internet of Medical Things (IoMT) aims to exploit the Internet-of-Things (IoT) techniques to provide better medical treatment scheme for patients with smart, automatic, timely, and emotion-aware clinical services. One of the IoMT instances is applying IoT techniques to sleep-aware smartphones or wearable devices' applications to provide better sleep healthcare services. As we all know, sleep is vital to our daily health. What is more, studies have shown a strong relationship between sleep difficulties and various diseases such as COVID-19. Therefore, leveraging IoT techniques to develop a longer lifetime sleep healthcare IoMT system, with a tradeoff between data transferring/processing speed and battery energy efficiency, to provide longer time services for bad sleep condition persons, especially the COVID-19 patients or survivors, is a meaningful research topic. In this study, we propose an IoT-enabled sleep data fusion networks (SDFN) module with a star topology Bluetooth network to fuse data of sleep-aware applications. A machine learning model is built to detect sleep events through an audio signal. We design two data reprocessing mechanisms running on our IoT devices to alleviate the data jam problem and save the IoT devices' battery energy. The experiments manifest that the presented module and mechanisms can save the energy of the system and alleviate the data jam problem of the device.**

***Index Terms*—Bluetooth, COVID-19, data fusion, Internet of Medical Things (IoMT), sleep healthcare, sleep-aware mobile application.**

Manuscript received November 16, 2020; revised January 26, 2021 and February 19, 2021; accepted March 17, 2021. Date of publication March 22, 2021; date of current version October 22, 2021. This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFA0706200; in part by the National Natural Science Foundation of China under Grant 61632014, Grant 61627808, and Grant 61210010; in part by the National Basic Research Program of China (973 Program) under Grant 2014CB744600; in part by the Program of Beijing Municipal Science & Technology Commission under Grant Z171100000117005; in part by the Science and Technology planning project of Shenzhen Municipality under Grant JCYJ20170818111012390 and Grant JCYJ20190807145209306; in part by the Shenzhen health committee under Project SZXJ2017034; and in part by the Shenzhen Key Medical Discipline Construction Fund under Grant SZXK005 and Grant SZXK012. *(Fan Yang and Qilu Wu are co-first authors.) (Corresponding author: Xiping Hu.)*

Fan Yang is with the School of Information Science and Engineering, Lanzhou University, Lanzhou 730000, China, and also with the the School of Mechanical & Automotive Engineering, South China University of Technology, Guangzhou 510006, China (e-mail: maxgundam@hotmail.com).

Qilu Wu, Rong Ma, and Bin Hu are with the School of Information Science and Engineering, Lanzhou University, Lanzhou 730000, China (e-mail: wuql20@lzu.edu.cn; marong@lzu.edu.cn; bh@lzu.edu.cn).

Xiping Hu is with the School of Information Science and Engineering, Lanzhou University, Lanzhou 730000, China, and also with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China (e-mail: huxp@lzu.edu.cn).

Jiancong Ye is with the South China University of Technology, Guangzhou 510006, China (e-mail: bf13760801483@gmail.com; haocongrao@gmail.com).

Yuting Yang is with the School of Electronics and Information Technology (School of Microelectronics), Sun Yat-sen University, Guangzhou 511400, China (e-mail: yangyt59@mail2.sysu.edu.cn).

Haocong Rao is with the School of Information Science and Engineering, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China.

Digital Object Identifier 10.1109/JIOT.2021.3067905

I. Introduction

The QUALITY of people's life is increasingly improved and the demand for medical resources is increasing day by day, which stand out the prominent disadvantages of the traditional medical pattern. At present, the main contradiction of the traditional pattern focuses on the following aspects: intensive medical resources, escalating conflicts between doctors and patients, unequal distribution of medical resources, etc. The Internet of Medical Things (loMT) emerges with the constant renewal and the development of a large number of portable sensors and integrated circuit processing units [1]. It provides a new way for the healthcare industry and eases the problems, such as the intensive medical resources and unequal distribution of medical resources. Besides, IoMT dramatically reduces the impact of artificial errors and medical errors on patients, according to a study at Johns Hopkins University [2]. Therefore, the research on IoMT is rather significant.

Defined as Vishnu *et al.* [3], the IoMT platform is an intelligent system, mainly obtaining the biomedical signal's sensor and the electronic circuits of patients, dealing with the processing units of biomedical signal, and storing the units through the network devices that transfer the biochemical data through the Internet. It is also convenient for doctors to make decisions according to the specific conditions of patients as an artificially intelligent visualization platform. The following are its four main applications.

1) *Diagnosis:* The IoMT devices track a growing number of physical indicators that can indicate some medical conditions, such as diabetes and atrial fibrillation, etc. Besides, it can be used to detect early signs of diseases or activities to discover possible diseases in time [4].
2) *Convalescence:* Postoperative recovery time is an important par of the operation cost, while minimizing the operation time is an important factor to reduce the cost. The sensor can track various key indicators and remind the nursing staff to react timely. Combining with the remote medical system makes it easier to accelerate the recovery [5].
3) *Long-Term Nurse:* With the development of blood pressure, glucose levels, sweat, and even tear analysis, sensors that track body parameters are becoming more and more

© IEEE 2021. This article is free to access and download, along with rights for full text and data mining, re-use and analysis.

sophisticated. Therefore, in the process of the chronic nurse, adverse outcomes and prolonged recovery period can be avoided by ideally applying the measurement and monitor for Internet-of-Things (IoT) devices to it [6].

4) *Precaution:* Let patients take the initiative to use IoMT equipment when participating in guided exercises, to avoid physical health problems caused by bad habits [7].

To sum up, IoMT is a product that combines the IoT and healthcare field [8]. It can be used to track key medical parameters, such as blood chemistry, blood pressure, brain activity, pain levels, etc. It can also help detect the early signs of some diseases or activities to further improve the body conditions [9].

According to the definition of health from the World Health Organization (WHO), owning a good sleep quality is one of the most important signs of a healthy body. The total sleep time (TST) accounts for about a third of one's life, so high quality sleep is very important to maintain one's best health condition. Another survey from the WHO said, 27% people in the world have sleep problems. So it is necessary to study how to monitor and evaluate the sleep quality and improve sleep problems.

The COVID-19 disease [10], whose virus name is severe respiratory syndrome Coronavirus 2 (SARS-COV-2), is now ravaging greatly. The relevant studies have shown that one of the common squealea of COVID-19 is the sleep difficulties [11], [12].

Many noninvasive sleep monitoring devices equipped with sleep-aware applications, such as smartphones or wearable devices, are beginning to emerge. These noninvasive sleep monitoring devices may provide possible help for COVID-19 sequelae patients with potential sleep problems in the future.

Sleep-aware applications, such as EAST [13] and smart alarm [14], use the accelerometer sensor and audio recorder in the smartphone to sample the acceleration and voice data of sleep users in a bedroom.

Sampled data will be sent to remote servers via IEEE 802.11 WLAN [15] for further analysis and services providing. However, Wi-Fi's working nominal range is about 100 m [16], which is unnecessary since the smartphones are in the same bedroom. In contrast, IEEE 802.15 [17], that is, Bluetooth, has 10-m working nominal range [16] and is more battery energy efficient [18] than Wi-Fi [16] in our situation. What is more, wireless devices, such as smartphones or wearable devices, bear the battery energy exhaustion problem in multiple situations. Therefore, these devices can hardly afford the high battery energy cost of most sleep-aware applications.

To address the above problems, we propose an IoT-enabled sleep data fusion networks (SDFN) module with a star topology Bluetooth network to fuse data of sleep-aware applications based on our devised application protocol. In the star topology network, a center Bluetooth IoT device fuses sleep data generated from every IoT device node in a room. The center will send the fused data to the remote server through the Ethernet cable. We design two data preprocessing mechanisms, SPL-based audio data reducing mechanism and signal power-based audio data selection mechanism, running on our IoT devices to alleviate the data jam problem and save the IoT devices' battery energy. The proposed methods and mechanisms can help to save the energy of sleep monitoring devices; therefore, it can provide longer sleep healthcare services for people with sleep problems, especially the COVID-19 patients or survivors. Our six contributions are summarized as follows.

1) We propose a new model called data fusion-enabled multimodal sleep-data analysis system (DF-MSAS). Based on the iSmile platform, this new model can analyze sleep data and provide sleep services in a more battery energy-efficient way.
2) A new module called SDFN is designed to replace the original Wi-Fi wireless communication way with Bluetooth way for battery energy efficiency.
3) We build a machine learning model, called SleepDetCNN, to detect the sleep event, such as snoring and coughing, through the audio data. This model runs on the center and provides higher level information that serves our proposed data preprocessing mechanism.
4) Although the SDFN module can save nodes' energy, this approach may reduce the data transfer speed, which may cause serious data jam. In order to deal with this problem, we design an SPL-based audio data reducing mechanism.
5) The Internet network condition is hard to control. A bad network condition may cause data jam at the star center device. Therefore, we design a signal power-based audio data selection mechanism to solve this problem, releasing the center device's data uploading burden.
6) In the experimental part, we study the relation between the received signal strength indication (RSSI) and energy consuming speed and find the region of RSSI in which the Bluetooth one is more battery energy efficient than the Wi-Fi one. What is more, we exhibit our proposed data preprocessing mechanisms' effects on devices' energy saving and data jam alleviating.

The remainder of this article is organized as follows. Section II concludes related research studies about IoMT, sleep healthcare, and COVID-19. Section III proposes our new system model DF-MSAS as well as our new module SDFN and describes their primary functions. Section IV describes several techniques and algorithms to implement our model. Section V describes our sleep event detection model's architecture. Section VI proposes several data preprocessing mechanisms to alleviate the data jam problem and further improve the devices' battery energy efficiency. Section VII conducts experiments to show that our model is more battery energy efficient than the original one and exhibit 2 mechanisms' effects on our model. Finally, experiment results are concluded and analyzed in section VIII.

II. Related Work

IoMT has experienced several stages. Initially, IoMT is the information management system to meet the daily operation of medical institutions. Then, IoMT extends to the medical management and monitoring field [19], such as combining the bar code technology with the medical management system, realizing the medical equipment's dynamic management, and further optimizing the scientific allocation of medical resources [20]. The problems of equipment maintenance caused by the traditional monitoring system can also be avoided by IoMT.

Applying IoMT technology and developing the lightweight, the ultralow energy consumption sensor equipment alleviates the discomfort to patients caused by traditional wearable monitoring equipment and avoids battery replacement and charging problems from happening.

A. Internet of Medical Things

1) Medical Cloud Architecture: In recent years, the feasibility of the combination of the cloud computing platform and intelligent medical system has been widely discussed [21]. The medical cloud architecture is initially formed, and the centralized medical cloud storage has much more mature solutions. However, medical imaging resources storage systems still rely on local data center, and the investment in a complex local platform will significantly increase medical care costs [22]. To solve these limitations, the medical cloud platform is born. But at present, most of the cloud platforms are unit architectures. When the platform encounters a resource bottleneck, medical institutions can only build a professional cloud platform or use online storage services to store medical data. However, these methods not only fail to effectively solve the problem of efficient storage but also create other security problems. Therefore, Cao *et al.* [23] proposed an extensible multistream storage architecture based on the medical cloud and designed a system called Tri-SFRS to improve the traditional IoMT architecture. The system can adapt to different storage environments and different access modes of medical data. It provides the fault recovery service of medical storage resources with the minimum communication costs and monitors medical resources changes in real time. At the same time, it is of higher security and stability and reduces the processing delay of the resource request.

2) IoMT Security System: Improving the IoMT system's security is an important research topic. Every IoMT system's inspector faces various risks, from the theft of private data to life-threatening security vulnerabilities [24]. Therefore, it is very important to identify and solve these security problems in time. At present, there is still a lack of architecture analysis in IoT planning research. Therefore, Rauscher and Bauer [25] developed a security analysis approach to identify security vulnerabilities in the IoMT architecture, which consists of a standardized meta model and an IoT security framework. This security analysis approach bridges the gaps between the relevant sensors and IoT, further improving the security of IoMT security.

3) Specific Applications of IoMT: This section summarizes the application of IoMT on vocal cord diseases and sleep problems.

a) Vocal cord diseases: According to a survey, the prevalence of vocal cord disease among teachers in their life time in the United States is 57.7%, while other professions are only 28.8% [26]. There are lots of intelligent systems developed to detect vocal cord disease. These systems can only determine whether vocal cord disease exists or not without identifying the type of the disease [27]. Ali *et al.* [28] proposed a vocal cord detection medical system based on bandpass filters to simulate the human hearing mechanism, and then, deployed the system into the IoT-based smart cities and smart homes to detect and classify various kinds of vocal cord diseases. According to the experimental results, the system is accurate and reliable in assessing vocal cord diseases. The accuracy of classifying the types of vocal cord diseases is higher than 95%.

b) Sleep healthcare: Except for the applications in vocal cord diseases, IoMT technology has many applications in monitoring, evaluating, and improving sleep quality. For example, Cao *et al.* [29] proposed a contactless body movement recognition (CBMR) method to collect the channel state information data of body movement. The CBMR method utilizes two types of IoT devices, which act as the Wi-Fi signal source and receiver's roles, respectively. By sliding a window, the channel state information data are segmented, and then use the recurrent neural network (RNN) [30], [31] to learn the context information of segmented channel state information data. Finally, use the Softmax function [32] to classify the types of human body movements that occurred during sleep. This method can effectively reduce the time consuming caused by data preprocessing and manual extraction of features and has an average classifying accuracy of over 93.5% on the complex human movement data set. Therefore, compared with other traditional methods, it can better identify human movement types during sleep and achieve higher accuracy in evaluating sleep quality.

B. Sleep-Aware Applications

There are lots of studies on the sleep pattern in order to better analyze and evaluate the quality of sleep. Based on these researches, various kinds of sleep assisting applications [13], [14], [33], [34] for smartphones were developed. The application iSleep [34] makes use of the smartphone's built-in microphone to record the sound of the user during their sleep. Then, nonnoise frames' features will be extracted and sent to a lightweight decision tree-based algorithm to classify the events that are closely related to sleep quality, such as body movement, couch, and snore, and infer quantitative measures of sleep quality according to the Pittsburgh sleep quality index (PSQI)-based questionnaire [35]. Based on iSmile Platform [33], the application EAST [13] extracts the sound features and classify the corresponding events into cough, snore, and sleep talk as iSleep does. What is more, it records the 3-D accelerator data of the user by the accelerometer sensor. After noise elimination and sleep features extraction, a multivariate neural random forest model is used to predict the valence-arousal value [36] of the user for sleep tips recommendation. In the smart alarm application [14], a k-NN-based method is designed to receive a vector of the user context model (UCM) as input and predict the corresponding arousal-valence values. According to the predicted arousal-valence values of UCM, the closet alarm sound in the arousal-valence plane is regard as the recommended one.

III. System Model

A. Data Fusion-Enabled Multimodal Sleep-Data Analysis System

Our works are based on iSmile Platform [33], and a new module called SDFN is added obtaining a new model called

DF-MSAS whose framework is shown in Fig. 1. The DF-MSAS makes use of the built-in audio recorder and accelerometer sensor of the smartphone to sample users' sleep data, and then sleep data will be sent to the remote server directly from sampling smartphones for further function services. The main three functions of this application are sleep tips recommendation, smart alarm recommendation, and sleep quality scoring. We now explain how we model, process, and analyze raw sleep data to enable these functions.

1) Mood Prediction and Tips Recommendation: The accelerometer sensor in the smartphone samples data in three directions x, y, and z of Cartesian coordinates for detecting tiny vibration of bed during users' sleep time. The whole night acceleration data will be sent to remote server for moods prediction and tips recommendation. Data will be first segmented into frames for sleep features extraction. Then, the algorithm will separate the movement events from the noise according to the noise threshold. After that, the algorithm will extract the statistical features of acceleration data, such as root mean square (rms), variance (var), and mean (avg) and send these features to a low-pass filter to detect the movement events. The algorithm computes sleep features of movement events, such as TST, movement rate (MR), average movement amplitude (AMA), and average movement interval (AMI), which can be used to measure the quality of sleep through the night. A multivariate neural random forest model that takes sleep features as inputs is built to predict the moods in the form of arousal and valence coordinates of the arousal-valence model. Finally, we generate suggestion tips for users to have a better sleep quality according to sleep features and moods.

2) Alarm Recommendation: Fig. 2 shows an overview of the smart alarm sound recommendation system. We now briefly describe the alarms recommendation approach used in the DF-MSAS. Every alarm sound is represented in a 6-D feature space with a well-defined similarity function defined by formula (1). The components of this feature space are zero-crossing rate (ZCR), tonal type (TT), tempo (TP), low energy rate (LER), spectral centroid (SC), and unit power (UP), which can well describe the properties of the alarm sounds [37]. Therefore, every sound can be represented as a vector AFV = (ZCR, TT, TP, LER, SC, UP).

$$\begin{aligned}\mathrm{sim}\left(AFV_i, AFV_j\right) = 1 &- \frac{\left|TT_i - TT_j\right|}{6\max\left(TT_i, TT_j\right)} - \frac{\left|TP_i - TP_j\right|}{6\max\left(TP_i, TP_j\right)} \\ &- \frac{\left|ZCR_i - ZCR_j\right|}{6\max\left(ZCR_i, ZCR_j\right)} \\ &- \frac{\left|LER_i - LER_j\right|}{6\max\left(LER_i, LER_j\right)} \\ &- \frac{\left|SC_i - SC_j\right|}{6\max\left(SC_i, SC_j\right)} - \frac{\left|UP_i - UP_j\right|}{6\max\left(UP_i, UP_j\right)}.\end{aligned} \tag{1}$$

We first manually map four alarm sounds to the arousal-valence model space by assigning arousal-valence values. Then, the algorithm will map other alarm sounds in the alarm sound library according to the similarity of the feature space among all the already mapped alarm sounds.

We define UCM, which combines feature vector (FV), context vector (CV), and social vector (SV) to represent different users in different situations. In our situation, FV is set to be the sleep features of users. The CV contains users' emotional states, which are the arousal-valence values and the real-time weather. The SV mainly considers the users' social information, such as age, occupation and academic degree

$$\begin{aligned}&\mathrm{sim}\left(UCM_i, UCM_j\right) \\ &\quad = W_f \times \mathrm{sim}\left(FV_i, FV_j\right) + W_c \times \mathrm{sim}\left(CV_i, CV_j\right) \\ &\quad\quad + W_s \times \mathrm{sim}\left(SV_i, SV_j\right)\end{aligned} \tag{2}$$

$$\begin{aligned}&\mathrm{sim}\left(FV_i, FV_j\right) \\ &= 1 - \sqrt{\sum_s \left(FV_{i_s} - FV_{j_s}\right)\left(FV_{i_s} - FV_{j_s}\right)}\end{aligned} \tag{3}$$

$$\begin{aligned}&\mathrm{sim}\left(CV_i, CV_j\right) \\ &= 1 - \sqrt{\sum_s \left(CV_{i_s} - CV_{j_s}\right)\left(CV_{i_s} - CV_{j_s}\right)}\end{aligned} \tag{4}$$

$$\begin{aligned}&\mathrm{sim}\left(SV_i, SV_j\right) \\ &= W_{\mathrm{age}} \times \left(1 - \frac{\left|\mathrm{age}_i - \mathrm{age}_j\right|}{\max\left(\mathrm{age}_i, \mathrm{age}_j\right)}\right) \\ &\quad + W_{degree} \times \left(1 - \frac{\left|\mathrm{degree}_i - \mathrm{degree}_j\right|}{\max\left(\mathrm{degree}_i, \mathrm{degree}_j\right)}\right) \\ &\quad + W_{\mathrm{nationality}} \times \mathrm{sim}\left(\mathrm{Nationality}_i, \mathrm{Nationality}_j\right) \\ &\quad + W_{gender} \times \left(1 - \left(\mathrm{gender}_i \oplus \mathrm{gender}_j\right)\right) \\ &\quad + W_{\mathrm{ocupation}} \times \mathrm{sim}\left(\mathrm{Occupation}_i, \mathrm{Occupation}_j\right).\end{aligned} \tag{5}$$

The similarity function between two UCMs are described by formula (2)–(5), where W_{age}, W_{degree}, $W_{\mathrm{nationality}}$, W_{gender}, and $W_{\mathrm{occupation}}$ can be defined by surveys or initial experiments for different situations. The only constraint here is $W_{\mathrm{age}}+W_{\mathrm{degree}}+W_{\mathrm{nationality}}+W_{\mathrm{gender}}+W_{\mathrm{occupation}} = 1$. Now, we can compute every UCM's corresponding arousal-valence values according to the predicted alarm sounds' arousal-valence values of top k similar users' history UCM. In other words, we map every UCM to the arousal-valence model space according to the history of UCM.

During the alarm sound prediction phase, the algorithm regards the alarm sound with the closest arousal-valence values to the current user's UCM's arousal-valence values as the predict recommendation result for the user.

3) Sleep Audio Scoring: This function takes sleep audio as input and detects as well as scores the sleep talks, snoring, and coughing during night time. Same as the process of dealing with the accelerator data, we first frame data and separate events from noise, and then extract events' features to predict its classification. Finally, the algorithm counts each kind of event's appear times as well as computes their intensity to score each event type and gives the total score of sleep basing the PSQI.

B. Sleep Data Fusion Networks's Framework

As mentioned in Section III-A, after sampling the sleep data, the smartphone will directly send the data to the remote server

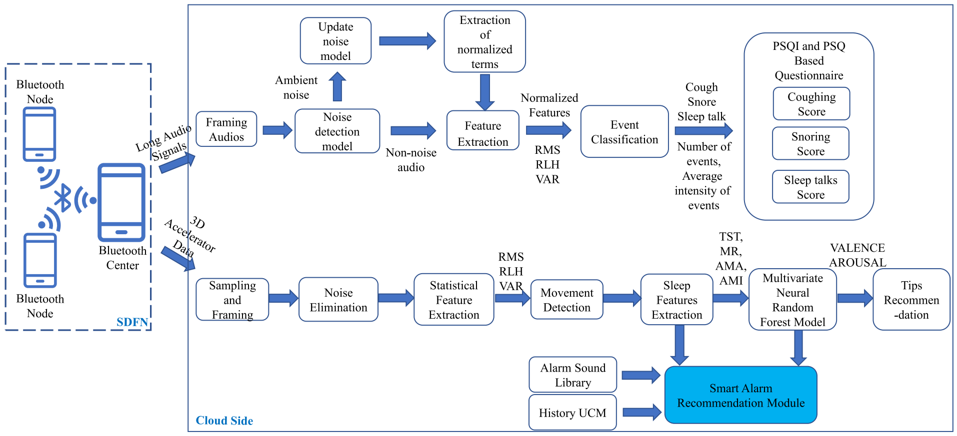


Fig. 1. DF-MSAS's framework.

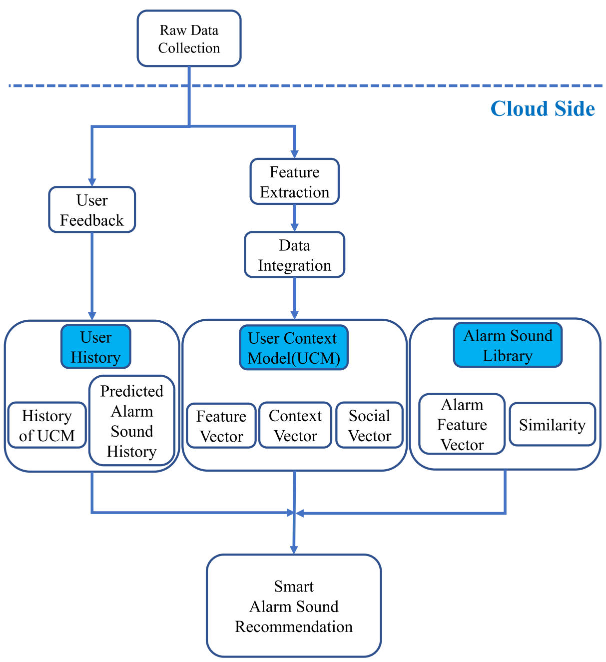


Fig. 2. Smart alarm sound recommendation module.

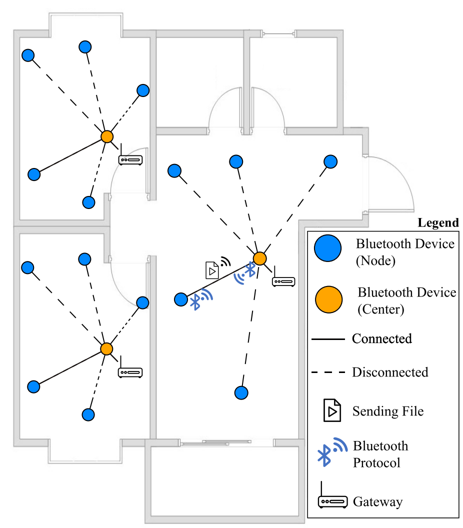


Fig. 3. Topology of Bluetooth network.

through Wi-Fi. However, compared to the Wi-Fi, Bluetooth is much more battery energy efficient. In order to improve the life time of every sampling smartphone, we first use a Bluetooth device center, which is connected to the power supply, to fuse the sleep data sampled by every sampling smartphone. Then, this device center will send fused data to the remote server. Our modification part in the iSmile framework, that is, SDFN, is shown in Fig. 1. The abridged general view of devices layout is shown in Fig. 3, and nodes in a star topology network are placed in the same bedroom for sampling sleep data of users via sleep application. The center actively connects to every node one by one through Bluetooth.

Once the connection is built, the node will send sampled data to center or receive commands for setting configures as well as operating node based on the protocol shown in Fig. 4. If $wflag = 1$, the center calls the readUTF() function, which blocks the center thread until the node calls function writeUTF() to send filename string to the center. Same as how center obtains filename string, center blocks until it has received file length. Then, the center receives the file itself and saves the file to disk according to the filename string. Finally, the node calls flush() to write the data to center from the buffer. As shown in Fig. 3, the center connects to the Internet through cable linking to gateway. The fused data in the center will be sent to the remote server via the Http protocol. If $wflag = 0$, center will send the command string to the node using writeUTF(), and the node will parse the command string and set configures subsequently. After setting configures, the node will call writeUTF() to send the feedback string to the center.

IV. Data Fusion Approach

A. Bluetooth Center

This section introduces the center's algorithms. As Algorithm 1 shows, center verifies every Bluetooth device in device list L, which is obtained by the Bluetooth discovery android API, whether it is target device or not by Algorithm

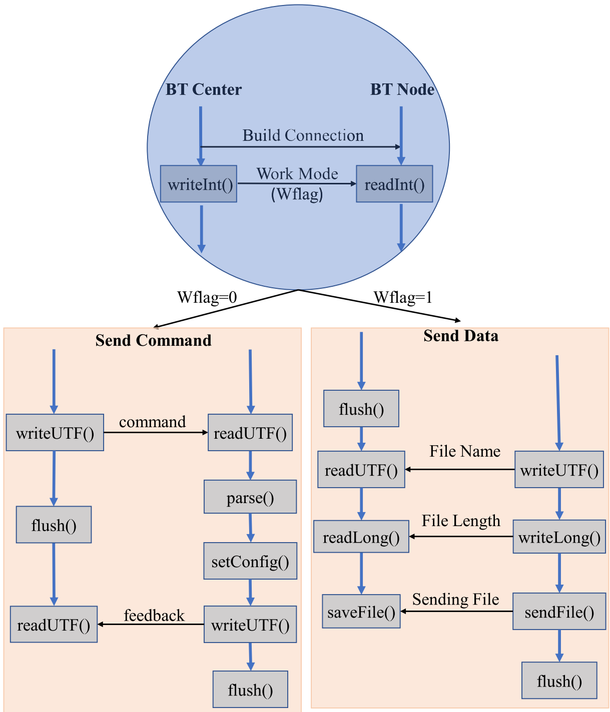


Fig. 4. Application layer protocol.

Algorithm 1 Data Fusing

Input: Bluetooth Devices List L
Device token string S
Fusing time interval T seconds

loop
  tmpFlag =*wflag*
  wflag = 1
  for node **in** L **do**
    if TargetDev(S,node.name) **then** {Only consider target device.}
      node.connect() {Connect Bluetooth device.}
      while not node.isConnected() **do**
        {Wait until connection is built.}
      end while
      if tmpFlag==1 **then**
        SaveFile() {Block until finishing receiving file.}
      else
        SendCMD()
      end if
    end if
  end for
  sleep(T) {Block thread T seconds before next turn.}
end loop

2. If it is the target device, then connect this node and ask for data or send command to the node for setting configures as well as the operating node based on the protocol in Section III. In the protocol, there is a *wflag* to differentiate between the data sending mode and command sending mode. *wflag* is set by another GUI thread. When finish scanning L, the center will sleep T seconds before the next turn.

Algorithm 2 TargetDev(S,N)

Input: Device token string S
Device name N
Output: Target Device Boolean Flag

for $i = 0$ **to** length(N) **do** {Decrypt device name.}
  if i is odd **then**
    swap($N[i-1], N[i]$) {Swap adjacent odd position character and even position character in string N }
  end if
end for
if isSubstring(S,N) **then** {Target device's real name string must contain S substring.}
  return true
else
  return false
end if

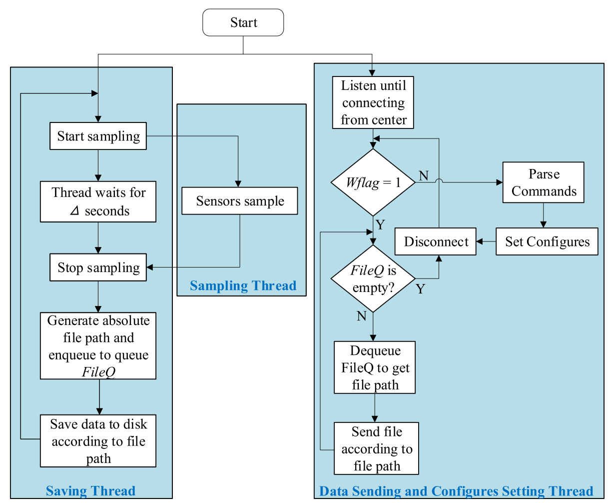


Fig. 5. Node's flowchart.

Algorithm 2 tells how to verify whether a device is a target device or not. First, decrypt the Bluetooth device's name string by swapping the adjacent odd position and even position character. If the device token string S, which is set manually by developers in advance, is the substring of decrypted name string, it is the target device.

B. Bluetooth Node

This section introduces the working flow of three subthreads in the Bluetooth node. As shown in Fig. 5, saving thread notifies the sampling thread to start sensors' sampling and then blocks for Δ seconds before stopping the sampling thread. This way Δ seconds of sensor data are generated. Saving thread generates filename according to the MAC address of node as well as the sampling time and enqueues this filename with absolute folder path into a queue termed *FileQ*. Finally, saving thread saves sleep data as file with the generated filename to disk according to the absolute folder path.

The data sending and configures setting thread will first judge *wflag*. If *wflag* = 1, dequeue *FileQ* obtains the filename from *FileQ*. After that, the thread will send the corresponding

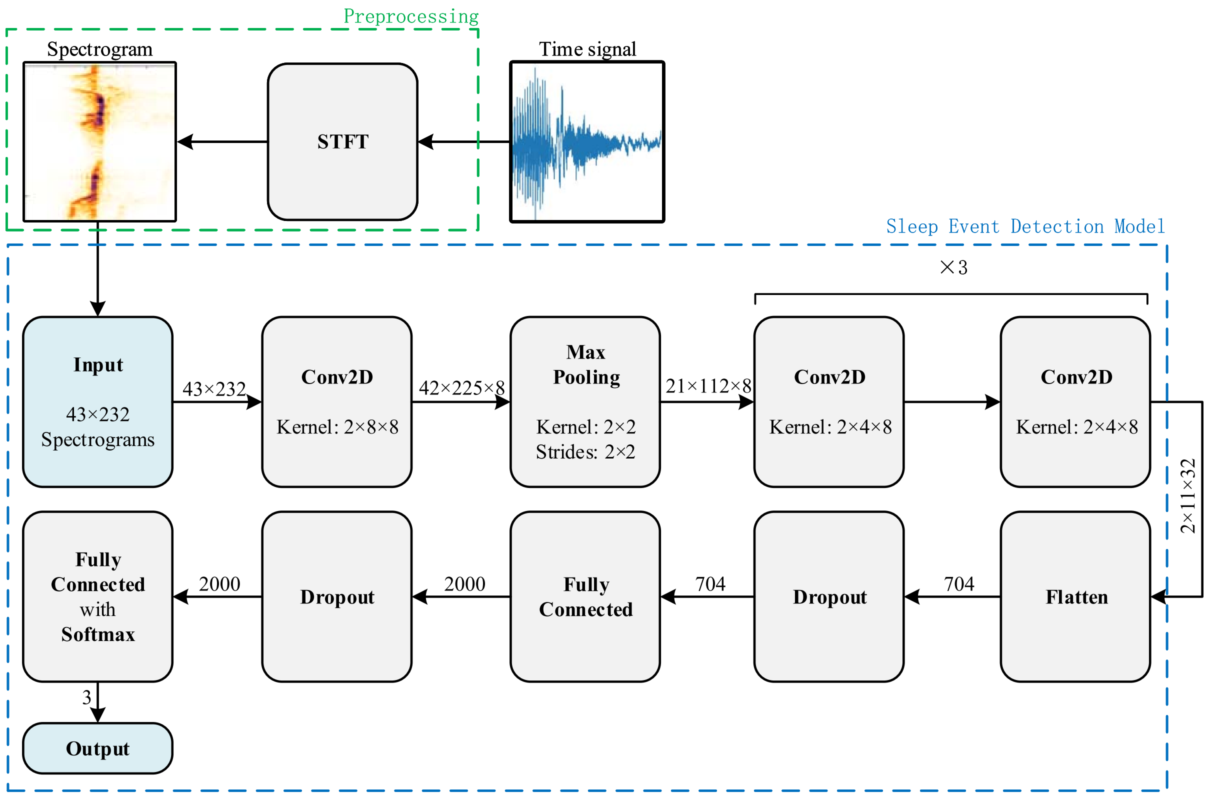


Fig. 6. Architecture of sleep event detection model.

Algorithm 3 Parse(S)

Input: Command String S

Output: Command List L

tmp = S.split("/") {Split the string to a list by character "/".}
L = {Initialize an empty list.}
for $i = 0$ **to** length(tmp) **do** {Generate Command List L.}
  L.append(tmp[i].split("#"))
end for
return L

file in the disk to the center based on the protocol mentioned aforehand. Finally, the file will be deleted from the disk when the center has received it. If $wflag = 0$, the thread will parse commands with following coding form by Algorithm 3:

"/command1#paramter1#paramter2/command2#paramter1$\cdots$"

V. Convolutional Neural Networks for Sleep Events Detection

We utilize a convolutional neural network (CNN) [32]-based sleep events detection model, called SleepDetCNN, to classify every time frame audio signal into three classes: 1) *Snoring*; 2) *Coughing*; and 3) *Other*. SleepDetCNN can provide higher than feature-level information [38] for advanced data preprocessing in Section VI-B.

Fig. 6 is the architecture of SleepDetCNN. In the inference phase of SleepDetCNN, a one second audio signal is first converted to a spectrogram by using the short-time Fourier transform (STFT) [39].

The STFT is a Fourier-related transform applied to a windowed time signal $x(n), n \in [-\infty, \infty]$. Formula (6) defines the STFT, where $W(n)$ is the window function, commonly a Hann window or Gaussian window centered around zero

$$\text{STFT}\{x(n)\}(t, \omega) = \sum_{n=-\infty}^{\infty} x(n)W(n-t)e^{-j\omega n}. \tag{6}$$

The spectrogram of signal $x(n)$ is the magnitude squared of the signal's STFT. Because the spectrogram is a order 2 tensor representation of the signal, it can be directly fed into a 2-D CNN model consisting of a 2-D convolutional layer, max-pooling layer, fully connected layer, and dropout layer [32].

The last fully connected layer's output will send to the Softmax function, which is defined by formula (7), to generate a 3-D probability distribution vector of three classes. The first, second, and third vector components refer to *Other*, *Snoring*, and *Coughing*, respectively. The class whose related vector component has the highest probability is the final classification result

$$\text{Softmax}(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad \text{for } i = 1, \ldots, K \quad \text{and } \mathbf{z} = (z_1, \ldots, z_K) \in \mathbb{R}^K. \tag{7}$$

In the CNN training phase, we prepare 798 audio files, each audio file with a 1-s duration, and each class has 266 audios. Most of the audios are extracted from the videos in

Audio Set [40]. All audios' spectrograms, together with the classification label, generate our data set.

The training target function is the cross-entropy loss function. We use the Adam optimizer [41] with 200 batch size each iteration to minimize the target function.

The minimum validation loss is found to determine the best training epoch, that is, 72, and the related validation accuracy is 79.32%.

We use the Keras deep learning framework to implement the spectrogram calculation process and the CNN model and combine them to generate a whole model. In the SleepDetCNN model deployment phase, the combined model is converted to the Tensorflow Lite model for android device deployment.

The SleepDetCNN model is running on the center, and whenever the center receives an audio file from the node, it will first pad the audio time duration to an integer by the node microphone's noise and then separate the whole audio into several 1-s audio files. After separation of the padded audio, every audio segment will be fed into the model to obtain the probability distribution vector. The average probability distribution vector over all vectors is regarded as the final result of SleepDetCNN over the original whole audio.

VI. Data Preprocessing Mechanisms of the IoT Devices

The computing capability and storage capacity of IoT devices are critical resources in our system. How to combine the resources in the IoT devices with that of the remote servers to reduce the overall battery energy consumption of the IoT devices, and to alleviate the data jam problem, and as a result, to provide better sleep health monitoring as well as evaluation services, is an important research topic [42].

In this section, we propose two mechanisms, the SPL-based audio data reducing mechanism and signal power-based audio data selecting mechanism, to utilize the IoT devices' resources to preprocess the sleep data. The mechanisms can dynamically fit the Bluetooth network's status and the Internet condition and keep the data transferring path's unimpeded.

A. SPL-Based Audio Data Reducing Mechanism

The aforementioned proposed approach uses a star topology Bluetooth network to fuse and upload sleep data generated from nodes in a room, instead of directly uploading the data via Wi-Fi. This approach may save every node's battery energy, but slows down the data transferring speed, which would cause data jam. Therefore, we propose the SPL-based audio data reducing mechanism to alleviate this problem.

1) Audio Data Compare Function: When the number of nodes in a room increases, the sleep data amount will largely increase, which will bring an enormous burden to the star topology Bluetooth network. In order to increase every center's capability of holding the nodes, it is necessary to design mechanisms to reduce the data amount generated from every node, especially the amount of audio data. As we all know, not all the time moment will the room has voices during night time. Therefore, we can design a mechanism to determine whether to record the audio data or not according to the sound pressure level (SPL) [43] defined by

$$\mathrm{SPL}(t) = 20\log_{10}\left(\frac{p(t)}{p_{\mathrm{ref}}}\right) \tag{8}$$

where $p(t)$ refers to the actual sound pressure (in Pa) [43] at time t and p_{ref} refers to the reference sound pressure, which is 20μPa [43].

We denote the digital output value of the microphone in the node device with i ID as $A^i(t)$. According to the sensor theory, the sound pressure $p(t)$ has a relation to $A^i(t)$ as follows [44]:

$$p^i(t) = k^i \times A^i(t) \tag{9}$$

where k^i is a parameter larger than 0 and varies with the hardware device. Different devices will have different microphones, with different sensitivities, and different audio hardware preamplifier. What is more, they may apply different input gain, equalizer, and automatic gain control (AGC) and noise reduction algorithms, explaining the variation of k^i.

Therefore, we have the formula (10) to compute SPL of node i at time t in the android programs

$$\begin{aligned}\mathrm{SPL}^i(t) &= 20\log_{10}\left(\frac{p^i(t)}{p_{\mathrm{ref}}}\right)\\ &= 20\log_{10}\left[\frac{k^i \times A^i(t)}{p_{\mathrm{ref}}}\right]\\ &= 20\left[\log_{10}k^i + \log_{10}A^i(t) - \log_{10}p_{\mathrm{ref}}\right].\end{aligned} \tag{10}$$

An audio data segment of a discrete time interval T_n from node i can be defined as a set of SPL at every time moment in T_n, that is, $\{\mathrm{SPL}^i(t)|t \in T_n\}$. For convenience, we denote $\{\mathrm{SPL}^i(t)|t \in T_n\}$ as $W^i_{T_n}$.

Then, we can define formula (11) to compare the SPL of two different audio data segments $W^i_{T_n}$ and $W^j_{T_m}$ at time interval T_n and T_m and from nodes i and j, respectively

$$\begin{aligned}&\mathrm{Compare}\left(W^i_{T_n}, W^j_{T_m}\right)\\ &\quad= \max\left(\{\mathrm{SPL}^i(t)|t \in T_n\}\right) - \max\left(\{\mathrm{SPL}^j(t)|t \in T_m\}\right)\\ &\quad= 20\left[\log_{10}k^i + \max\left(\{A^i(t)|t \in T_n\}\right) - \log_{10}p_{\mathrm{ref}}\right]\\ &\qquad-20\left[\log_{10}k^j + \max\left(\{A^j(t)|t \in T_m\}\right) - \log_{10}p_{\mathrm{ref}}\right]\\ &\quad= \max\left(\{A^i(t)|t \in T_n\}\right) - \max\left(\{A^j(t)|t \in T_m\}\right)\\ &\qquad+\log_{10}k^i/k^j\end{aligned} \tag{11}$$

Especially, when node ID $i = j$, we have formula (12), which only relates to the digital output value $A^i(t)$

$$\begin{aligned}&\mathrm{Compare}\left(W^i_{T_n}, W^i_{T_m}\right)\\ &\quad= \max\left(\{A^i(t)|t \in T_n\}\right) - \max\left(\{A^i(t)|t \in T_m\}\right).\end{aligned} \tag{12}$$

2) Audio Data Reducing Mechanism: In the previous section, we define a *Compare*$(W^i_{T_n}, W^j_{T_m})$ function that can judge which audio data segment contains a louder voice. With this function's help, we can design a mechanism for recording the audio, only when there are voices in the environment.

Before the node starts sampling the sleep data, we first place the center and node into an environment with a slight white noise [45]. Then, use the center to set node's SPL threshold W^i_0 by sending the *set threshold* command from the center to node i.

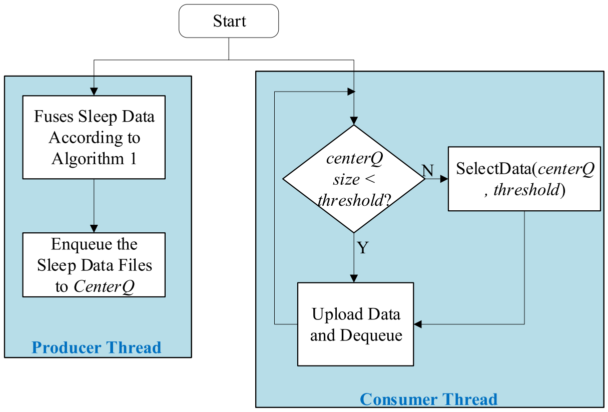


Fig. 7. Center's flowchart.

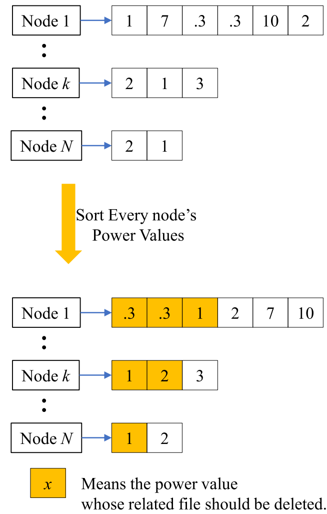


Fig. 8. Data selection mechanism instance.

After setting the threshold W_0^i of node i, when the sampling starts, node i will record the audio data only when $Compare(W_{T_n}^i, W_0^i) > 0$.

If $Compare(W_{T_n}^i, W_0^i) \leq 0$, then the node i will record the $\max(\{A^i(t)|t \in T_n\})$ value by saving it to a text file.

B. Signal Power-Based Audio Data Selection Mechanism

During the uploading of fused sleep data from the center to the remote server, network problems, such as network congestion [46], etc., may occur because a large amount of data will be uploaded at the same time from the centers in different geographic regions. When network problems occur, the sleep data will jam at the center. After the network problems are solved, and the network recovers to the normal condition, the center may already accumulate a large amount of sleep data, so it is difficult for the center to upload all accumulated sleep data at once. What is more, the center device may be a storage capacity restricted hardware. If the sleep data size exceeds the center device's maximum storage capacity, the center may crash, which is a hazard to our whole system. Therefore, it is necessary to design a selection mechanism that can retain more important data in the meaning of the defined indicator and discard less important data.

In this section, we first define an audio signal's information measurement in a finite discrete time interval and then describe how our proposed signal power-based audio data selection mechanism works in detail.

1) Indicator of Data Selection Mechanism: This section describes the indicator used to measure an audio signal's information quantity over a finite discrete time interval. A indicator consists of two components: 1) the sleep event class number i^* and 2) the relative average power p.

Denote the SleepDetCNN computed probability distribution vector of the audio file as $\vec{v} = (\vec{v}_1, \vec{v}_2, \vec{v}_3)$, and the sleep event class number i^* of the audio file is defined by

$$i^* = \underset{i}{\operatorname{argmax}}\, \vec{v}_i, \;\; i = 1, 2, 3. \tag{13}$$

According to the signal processing theory [47], the average power over a time interval of a finite length discrete digital time signal can be defined by

$$\frac{1}{N}\sum_{n=0}^{N}|x(n)|^2. \tag{14}$$

In our case, $x(n)$ means the audio signal recorded by microphone, that is, the sound pressure $p(t)$. Therefore, we have formula (15) to compute the energy of an audio signal set of discrete time interval T_n from node i, where the mathematical operation $card()$ means the cardinality of a set

$$\begin{aligned}\text{AveragePower}(\{p(t)|t \in T_n\}) &= \frac{1}{\text{card}(T_n)}\sum_{t\in T_n}|p(t)|^2 \\ &= \frac{1}{\text{card}(T_n)}\sum_{t\in T_n}\left[k^i \times A^i(t)\right]^2.\end{aligned} \tag{15}$$

Suppose we only compare the audio data from the same node. In that case, we can directly set the parameter k^i to 1 in formula (15), and now we obtain our second indicator component, relative average power p.

The tuple of two components (i^*, p) is regarded as the indicator, which fuses the event-level information i^* and feature-level information p for measuring an audio signal's information quantity.

2) Data Selection Mechanism: The center's program implementation uses a producer-consumer multithreads model [48]. According to Algorithm 1, the producer thread fuses the data generated from nodes in a room via the star topology Bluetooth network. Then, as Fig. 7 describes, enqueues saved files' path strings to a center file queue, denoted as centerQ. On the other hand, the consumer thread plays the role of an HTTP uploader as well as the data selection mechanism executor. Fig. 7 shows the center's flowchart. The consumer thread first judges whether the centerQ's size is smaller than the predetermined threshold. If true, the consumer thread uploads all data in the centerQ and dequeue as well as delete successfully uploaded data files. If false, the consumer thread calls the SelectData function defined by Algorithm 4 to enable the data selection mechanism.

Algorithm 4 SelectData(*centerQ*, *T*)

Input: files queue on center *centerQ*
predetermined queue size threshold T

Create empty array L and empty queue P.
for $i = 0$ **to** length(*centerQ*) **do** {Compute every audio files' average power, and save the power value as well as it's index in *centerQ* to the L.}
  if *centerQ*[*i*] is audio file **then**
    p = *RelativeAveragePower*(*centerQ*[*i*]) {Compute average power of current audio file.}
    i^* = SleepDetCNN(*centerQ*[i])
    $i^* = \text{argmax}_j\, i^*_j$ {Compute the sleep event class number of current audio file.}
    L.append((i, i^*, p))
  end if
end for
Sort(L, "ascendingly") {Sort the L ascendingly according to every element's class number i^*, then the elements with the same class number will be further sorted ascendingly by the p.}
Create a dictionary D according to T, which associates every node's ID to every node's number of files to be deleted.
for $i = 0$ **to** length(L) **do** {Delete less important audio files in *centerQ*.}
  $j = L[i][0]$
  if *centerQ*[*j*] is audio file **then**
    ID = NodeID(*centerQ*[*j*]) {Get the node ID that the file belongs to.}
    if $D[ID] > 0$ **then**
      $D[ID] = D[ID] - 1$
      DeleteFile(*centerQ*[*j*])
    else
      P.append(*centerQ*[*j*])
    end if
  else
    P.append(*centerQ*[*j*])
  end if
end for
centerQ = P

Algorithm 4 describes our signal power-based audio data selection mechanism and Fig. 8 shows an example of how the selection mechanism works.

In order to create a dictionary, which associates every node's ID to every node's number of files to be deleted, we need to first compute the reducing proportion Q of the centerQ according to formula (16). Then, every node's number of files in the centerQ times the proportion Q, and we obtain every node's number of files to be deleted

$$Q = \begin{cases} \frac{\text{centerQSize}-\text{threshold}}{\text{centerQSize}}, & \text{centerQSize} > \text{threshold} \\ 0, & \text{centerQSize} \leq \text{threshold}. \end{cases} \quad (16)$$

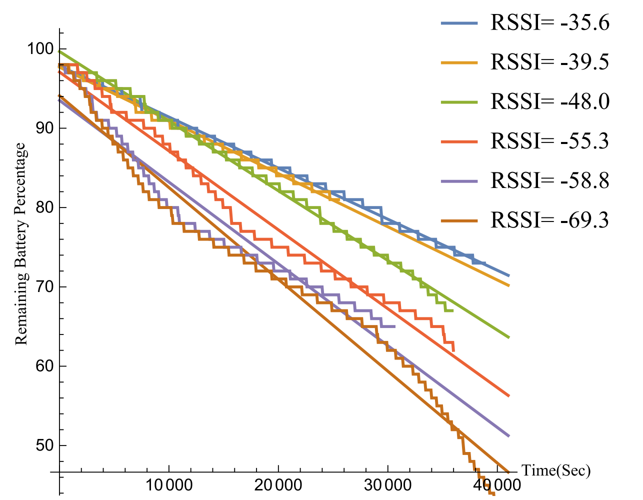


Fig. 9. Wi-Fi one's remaining battery percentage-time curve with different RSSI.

VII. Experiment

A. Experiment for Proposed Module SDFN

In this section, we aim at answering the following questions by conducting experiments.

1) What is the relation between power-consuming speed and RSSI?
2) When is the Bluetooth version app's power-consuming speed lower than that of the Wi-Fi one?

We denote the remaining battery percentage of time t as $R(t)$, and then the power-consuming speed K can be defined by formula (17), where $\hat{R}(t)$ means the least square regression line of $R(t)$

$$K = \left| \frac{\mathrm{d}\hat{R}(t)}{\mathrm{d}t} \right|. \quad (17)$$

To find the relation between K and RSSI, we ran our application in Meizu Note 5 and recorded the remaining battery percentage changing through time in different RSSI with the unit of dBm. The tools to measure the RSSI of Wi-Fi and Bluetooth are Ce Wang Su[1] and Bluetooth RSSI App android applications, respectively. Then, we draw the results of the Wi-Fi version and Bluetooth version in Figs. 9 and 10, respectively.

As we can see from Figs. 9 and 10, the power-consuming speed of the Wi-Fi one is much more sensitive to the RSSI than that of Bluetooth one, which means that there must be a critical point K_0, which discriminates the regions where $K_{\text{Bluetooth}} > K_{Wi-Fi}$ as well as $K_{\text{Bluetooth}} < K_{Wi-Fi}$.

To find K_0, we compute the regression line of $K_{\text{Bluetooth}}$ and K_{Wi-Fi} with respect to RSSI and obtain Fig. 11. We find that $K_0 = -39.45$ (dBm), which means that when RSSI < -39.45 (dBm), Bluetooth version is more battery energy efficient than Wi-Fi version.

However, the smaller the Bluetooth's RSSI is, the lower the transport speed the Bluetooth has. If the transfer speed is

[1] A Chinese android Wi-Fi RSSI measuring application, whose name is measure network speed in English.

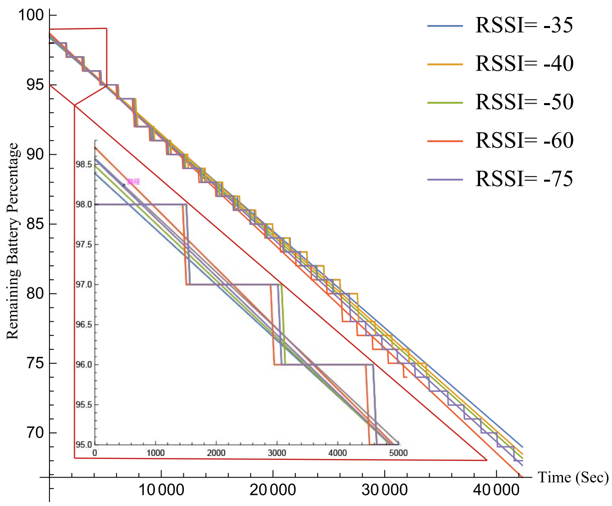


Fig. 10. Bluetooth one's remaining battery percentage-time curve with different RSSI.

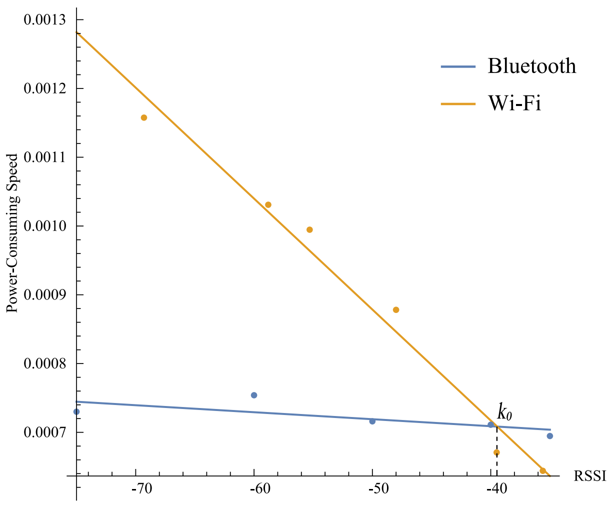


Fig. 11. Relation between power-consuming speed and RSSI.

lower than sleep data generating speed, there will be a time gap between the latest generated data and the latest transferred data. We term this time gap divided by total data sampling time as the relative delay time (RDT). The quantitative relationship between RDT and RSSI, which are described as a linear regression line, is shown in Fig. 12. We set the RDT threshold to 0.1, which means that supposing the total sampling time is 10 h and then the maximum acceptable time gap is 1 h. According to the relation between RDT and RSSI, we have a minimum of RSSI $= -59.29$.

Finally, we obtain an RSSI interval $[-59.29, -39.45]$. When the Bluetooth RSSI is in this interval, our new module SDFN can make our system more battery energy efficient without exceeding the delay-time threshold.

B. Experiment for Proposed Data Preprocessing Mechanisms

In Section VII-A, the experimental object, smartphone Meizu Note 5, has both a Wi-Fi module and a Bluetooth module as the data communication module. In this section,

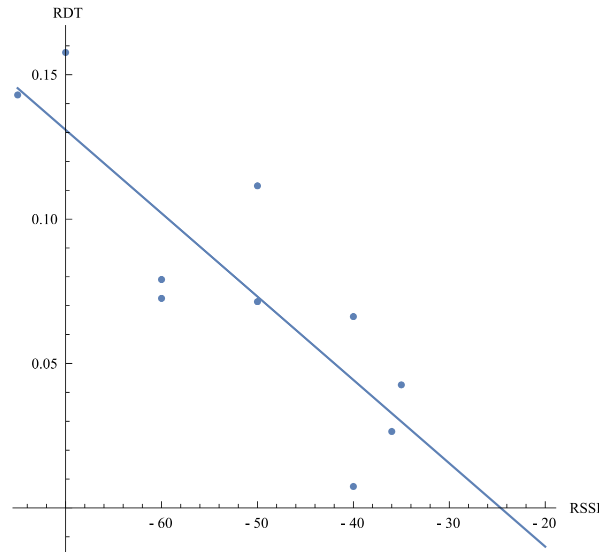


Fig. 12. Relationship between RDT and RSSI.

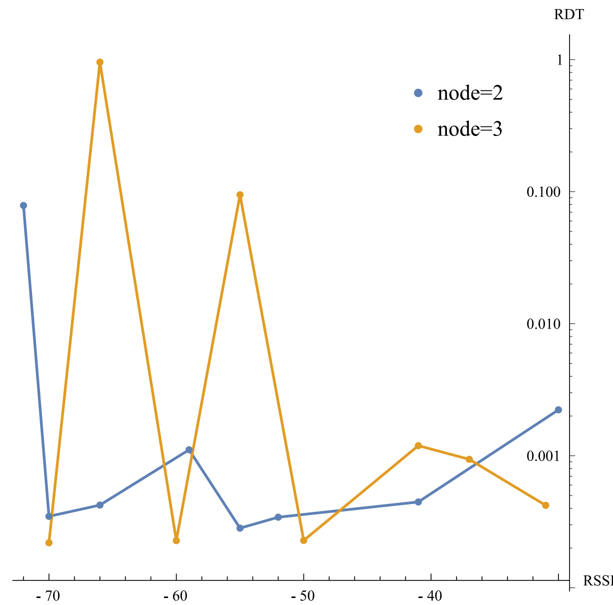


Fig. 13. RDT – RSSI curves in different node numbers.

we use our android IoT device instead. The IoT device, which plays the node's role, only has a Bluetooth module as the data communication module for battery energy efficiency.

This section aims to show our proposed mechanisms' effects on our android IoT devices' battery energy saving and data consuming speed. We conduct an experiment to show the effect on reducing audio data in *AAC* file format when applying the SPL-based data reducing mechanism to the SDFN module. The experimental results are shown in Fig. 13. According to the figure, almost all RDT results are under 0.010 in different RSSI and different node number settings. In contrast, as Fig. 12 shows, without applying the SPL-based data reducing mechanism, the RDT will larger than 0.10 when RSSI is less than about -60 dBm. Therefore, this mechanism can decrease the RDT. In other words, it can alleviate the

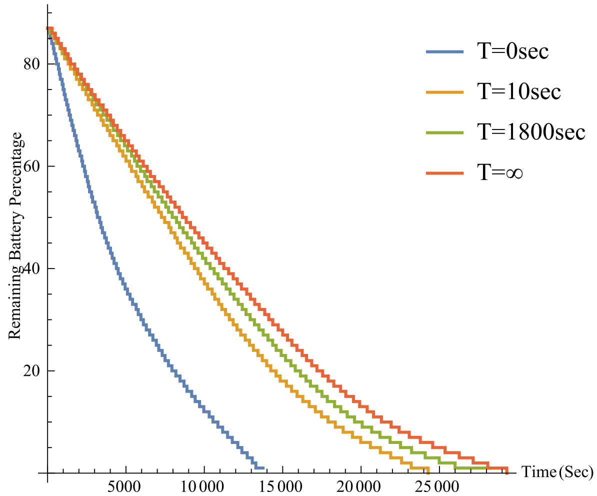


Fig. 14. Battery-consuming speed comparison experiment with or without Bluetooth data transferring.

data jam problem during the data transferring via Bluetooth network.

What is more, the SPL-based data reducing mechanism can help further saving the node's battery energy. As illustrated in Fig. 14, the battery-consuming speed is faster when center's sleep time interval T becomes shorter. Especially, when close Bluetooth data transferring function, that is, $T = \infty$, the IoT device has the longest battery life time. Therefore, the less Bluetooth data transferring time the node has during the data sampling process, the slower the battery consuming speed is, which means it is more battery energy efficient. The mechanism can significantly reduce the total data size, which means less data transferring time when the transferring speed is the same. As the result, this mechanism can save the battery energy of the node.

VIII. Conclusions and Future Work

In this article, we summarized the implementation methods of iSmile Platform-based android sleep-aware applications EAST and smart Alarm. Then, we proposed a module called SDFN, which uses the Bluetooth protocol instead of Wi-Fi to fuse sleep data basing our designed application protocol. Combining this new module with EAST and smart alarm, we obtained our new model DF-MSAS. A CNN-based sleep event detection model, SleepDetNet, is built, which makes use of the spectrogram of audio signal and the machine learning technique to classify the sleep event of every 1-s audio signal. We proposed the SPL-based audio data reducing mechanism and the signal power-based audio data selection mechanism to alleviate the data jam problem and further save the IoT devices' battery energy. We experimented and found that when RSSI $\in [-59.29, -39.45]$, DF-MSAS is more battery energy efficient with bearable data delay time. What is more, further experiments show that the proposed two mechanisms can alleviate the data jam problem and further save the IoT devices' battery energy.

In the future, we plan to design an abstract conception model and several interfaces to fit different IoT projects and make use of edge computing techniques and artificial intelligence algorithms to further reduce data delay time and improve the IoT device's battery energy efficiency.

References

[1] R. P. Singh, M. Javaid, A. Haleem, R. Vaishya, and S. Ali, "Internet of Medical Things (IoMT) for orthopaedic in covid-19 pandemic: Roles, challenges, and applications," *J. Clin. Orthopaedics Trauma*, vol. 11, no. 4, pp. 713–717, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S097656622030179X

[2] M. A. Makary and M. Daniel, "Medical error—The third leading cause of death in the U.S.," *BMJ*, vol. 353, May 2016, Art. no. i2139.

[3] S. Vishnu, S. J. Ramson, and R. Jegan, "Internet of Medical Things (IoMT)—An overview," in *Proc. IEEE 5th Int. Conf. Devices Circuits Syst. (ICDCS)*, 2020, pp. 101–104.

[4] T. J. Jebaseeli, C. A. D. Durai, and J. D. Peter, "IoT based sustainable diabetic retinopathy diagnosis system," *Sustain. Comput. Informat. Syst.*, vol. 28, Dec. 2020, Art. no. 100272. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2210537918301872

[5] D. Mišić, M. Zdravković, M. Mitković, N. Vitković, and M. Mitković, "Real-time monitoring of bone fracture recovery by using aware, sensing, smart, and active orthopedic devices," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4466–4473, Dec. 2018.

[6] R. Basatneh, B. Najafi, and D. G. Armstrong, "Health sensors, smart home devices, and the Internet of Medical Things: An opportunity for dramatic improvement in care for the lower extremity complications of diabetes," *J. Diabetes Sci. Technol.*, vol. 12, no. 3, pp. 577–586, 2018.

[7] B. Karthik, L. D. Parameswari, R. Harshini, and A. Akshaya, "Survey on IoT & arduino based patient health monitoring system," *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, vol. 3, no. 1, pp. 1414–1417, 2018.

[8] W. Zhao, C. Wang, and Y. Nakahira, "Medical application on Internet of Things," in *IET International Conference on Communication Technology and Application (ICCTA)*, Jan. 2011, pp. 660–665.

[9] A. Raji, P. G. Jeyasheeli, and T. Jenitha, "IoT based classification of vital signs data for chronic disease monitoring," in *Proc. IEEE 10th Int. Conf. Intell. Syst. Control (ISCO)*, 2016, pp. 1–5.

[10] C. Sohrabi *et al.*, "World health organization declares global emergency: A review of the 2019 novel coronavirus (COVID-19)," *Int. J. Surg.*, vol. 76, pp. 71–76, Apr. 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1743919120301977

[11] C. Huang *et al.*, "6-month consequences of COVID-19 in patients discharged from hospital: A cohort study," *Lancet*, vol. 397, no. 10270, pp. 220–232, 2021.

[12] M. Taquet, S. Luciano, J. R. Geddes, and P. J. Harrison, "Bidirectional associations between COVID-19 and psychiatric disorder: Retrospective cohort studies of 62 354 COVID-19 cases in the USA," *Lancet Psychiatry*, vol. 8, no. 12, pp. 130–140, 2021.

[13] Y. Guo *et al.*, "POSTER: Emotion-aware smart tips for healthy and happy sleep," in *Proc. 23rd Annu. Int. Conf. Mobile Comput. Netw.*, 2017, pp. 549–551.

[14] J. Wang *et al.*, "Mobile crowdsourcing based context-aware smart alarm sound for smart living," *Pervasive Mobile Comput.*, vol. 55, pp. 32–44, Apr. 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1574119218302104

[15] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Standard 802.11, Jun. 2007.

[16] J.-S. Lee, Y.-W. Su, and C.-C. Shen, "A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and Wi-Fi," in *Proc. Ind. Electron. Soc.*, vol. 5, 2007, pp. 46–51.

[17] J. T. Adams, "An introduction to IEEE std 802.15.4," in *Proc. IEEE Aerosp. Conf.*, 2006, p. 8.

[18] T. G. Xydis and S. Blake-Wilson, *Security Comparison: Bluetooth TM Communications vs. 802.11*, Bluetooth Security Experts Group, Kirkland, WA, USA, 2002.

[19] H.-Y. Zhu, P. Han, Y. Tian, Y. Wang, and Y.-C. Li, "Application of information technology in control and monitoring of nosocomial infections," *Chin. J. Nosocomiol.*, vol. 22, no. 17, pp. 3808–3810, 2012.

[20] L. Zhang and X.-B. Yang, "The application of bar code in hospital medical equipment management," *Chin. J. Med. Device*, vol. 5, no. 9, pp. 1–2, 2007.

[21] M. Kyrö, K.-I. Takizawa, K. Haneda, and P. Vainikainen, "Validation of statistical channel models for 60 GHz radio systems in hospital environments," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 5, pp. 1458–1462, May 2013.

[22] B. Xu, L. Da Xu, H. Cai, C. Xie, J. Hu, and F. Bu, "Ubiquitous data accessing method in IoT-based information system for emergency medical services," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1578–1586, May 2014.

[23] R. Cao, Z. Tang, C. Liu, and B. Veeravalli, "A scalable multicloud storage architecture for cloud-supported medical Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 3, pp. 1641–1654, Mar. 2020.

[24] M. Stanislav and T. Beardsley, "Hacking IoT: A case study on baby monitor exposures and vulnerabilities," in *Proc. Rapid7 Report*, 2015, pp. 68–78.

[25] J. Rauscher and B. Bauer, "Safety and security architecture analyses framework for the Internet of Things of medical devices," in *Proc. IEEE 20th Int. Conf. e-Health Netw. Appl. Services (Healthcom)*, 2018, pp. 1–3.

[26] N. Roy, R. M. Merrill, S. Thibeault, R. A. Parsa, S. D. Gray, and E. M. Smith, "Prevalence of voice disorders in teachers and the general population," *J. Speech Lang. Hearing Res.*, vol. 47, no. 1, pp. 281–293, Apr. 2004.

[27] Z. Ali, M. Talha, and M. Alsulaiman, "A practical approach: Design and implementation of a healthcare software for screening of dysphonic patients," *IEEE Access*, vol. 5, pp. 5844–5857, 2017.

[28] Z. Ali, M. S. Hossain, G. Muhammad, and A. K. Sangaiah, "An intelligent healthcare system for detection and classification to discriminate vocal fold disorders," *Future Gener. Comput. Syst.*, vol. 85, pp. 19–28, Aug. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X17324421

[29] Y. Cao *et al.*, "Contactless body movement recognition during sleep via WiFi signals," *IEEE Internet Things J.*, vol. 7, no. 3, pp. 2028–2037, Mar. 2020.

[30] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[31] S. Li, W. Li, C. Cook, C. Zhu, and Y. Gao, "Independently recurrent neural network (INDRNN): Building a longer and deeper RNN," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5457–5466.

[32] I. J. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning*, vol. 1. Cambridge, MA, USA: MIT Press, 2016.

[33] X. Hu *et al.*, "Emotion-aware cognitive system in multi-channel cognitive radio ad hoc networks," *IEEE Commun. Mag.*, vol. 56, no. 4, pp. 180–187, Apr. 2018.

[34] T. Hao, G. Xing, and G. Zhou, "iSleep: Unobtrusive sleep quality monitoring using smartphones," in *Proc. 11th ACM Conf. Embedded Netw. Sensor Syst.*, 2013, p. 4.

[35] 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," *Psychiat. Res.*, vol. 28, no. 2, pp. 193–213, 1989. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0165178189900474

[36] P. Bustamante, N. L. Celani, M. Perez, and O. Q. Montoya, "Recognition and regionalization of emotions in the arousal-valence plane," in *Proc. 37th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, 2015, pp. 6042–6045.

[37] X. Hu *et al.*, "SAFEDJ: A crowd-cloud codesign approach to situation-aware music delivery for drivers," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 12, no. 1s, p. 21, 2015.

[38] L. Wald, "Some terms of reference in data fusion," *IEEE Trans. Geosci. Remote Sens.*, vol. 37, no. 3, pp. 1190–1193, May 1999.

[39] N. Kehtarnavaz, *Digital Signal Processing System Design: LabVIEW-Based Hybrid Programming*. Amsterdam, The Netherlands: Elsevier, 2011.

[40] J. F. Gemmeke *et al.*, "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2017, pp. 776–780.

[41] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014. [Online]. Available: arXiv:1412.6980.

[42] J. Cao, Q. Zhang, and W. Shi, *Edge Computing: A Primer*. Heidelberg, Germany: Springer, 2018.

[43] D. M. Howard and J. Angus, *Acoustics and Psychoacoustics*, 2nd ed. New York, NY, USA: Focal, 2001.

[44] J. Fraden, *Handbook of Modern Sensors: Physics, Designs, and Applications*. New York, NY, USA: Springer, 2004.

[45] J. G. Proakis, *Digital Signal Processing: Principles Algorithms and Applications*. London, U.K.: Pearson Educ., 2001.

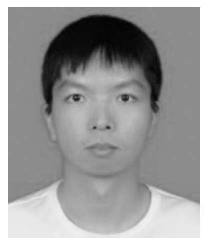
[46] A. H. Al-Bayatti and H. M. Al-Bayatti, *Simulation in Computer Network Design and Modeling: Use and Analysis*. Hershey, PA, USA: IGI Global, 2012.

[47] A. V. Oppenheim, A. S. Willsky, and S. H. Nawab, *Signals & Systems*, 2nd ed. Harlow, U.K.: Prentice-Hall, 1996.

[48] M. Herlihy and N. Shavit, *The Art of Multiprocessor Programming*. San Francisco, CA, USA: Morgan Kaufmann, 2008.

Fan Yang received the B.Eng. degree from China Jiliang University, Hangzhou, China, in 2018. He is currently pursuing the M.Eng. degree with the School of Mechanical and Automotive Engineering, South China University of Technology, Guangzhou, China.

He is also a Research Assistant with the School of Information Science and Engineering, Lanzhou University, Lanzhou, China. From 2019 to 2020, he was an internship student with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. His current research interests include pattern recognition, machine learning, and interpretable artificial intelligence.



Qilu Wu received the M.Eng. degree from Lanzhou University, Lanzhou, China, in 2020, where he is currently pursuing the Ph.D. degree with the School of Information Science and Engineering.

He is also a guest student with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. His current research interests include human–computer interactions and big data analyses.

Xiping Hu received the Ph.D. degree from the University of British Columbia, Vancouver, BC, Canada, in 2015.

He is currently a Professor with Lanzhou University, Lanzhou, China, and Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. He is also the Co-Founder and the Chief Scientist with Erudite Education Group Ltd., Hong Kong, a leading language learning mobile application company with over 100 million users, and listed as top two language education platform globally. His research includes mobile cyber–physical systems, crowdsensing, social networks, and cloud computing. He has more than 100 papers published and presented in prestigious conferences and journals, such as IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE INTERNET OF THINGS, ACM TOMM, IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, *IEEE Communications Magazine*, IEEE NETWORK, HICSS, ACM MobiCom, and WWW.

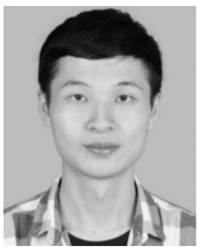
Prof. Hu has been serving as the Lead Guest Editor for IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING and WCMC.

Jiancong Ye received the B.E. degree from South China University of Technology, Guangzhou, China, in 2020, where he is currently pursuing the M.E. degree with the Shien-Ming Wu School of Intelligent Engineering.

From 2019 to 2020, he was an Algorithm Intern with Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. His current research interests include machine learning and artificial intelligence.

Yuting Yang received the B.S. degree from South China University of Technology, Guangzhou, China, in 2020. She is currently pursuing the M.S. degree with the School of Electronics and Information Technology (School of Microelectronics), Sun Yat-sen University, Guangzhou.

From 2019 to 2020, she was an Algorithm Intern with the Shenzhen Institute of Advanced Technology Chinese Academy of Sciences, Shenzhen, China. Her current research interest is Internet of things.

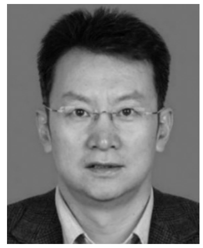


Haocong Rao received the B.E. degree from South China University of Technology, China, in 2019. He is currently pursuing the Ph.D. degree with Nanyang Technological University, Singapore.

He is currently a visiting student with Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. His research interests include pattern recognition, machine learning, and interpretable artificial intelligence, with current focus on person reidentification, action recognition, and unsupervised deep learning.

Rong Ma received the B.S., M.S., and Ph.D. degrees from Lanzhou University, Lanzhou, China, in 2004, 2007, and 2017, respectively.

She is currently an Engineer with Lanzhou University. Her current research interests include digital image processing, computational vision, and edge computing.



Bin Hu (Senior Member, IEEE) received the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Science, Beijing, China, in 1998.

From 2009 to 2019, he served as the Dean of the School of Information Science and Engineering, Lanzhou University, Lanzhou, China. His current research interests include pervasive computing, cognitive computing, and mental health care.

Dr. Hu served as an Editor for *IET Communications*, *Cluster Computing*, *Wireless Communications and Mobile Computing*, *Journal of Internet Technology*, *Security and Communication Networks* (Wiley), and *Brain Informatics*, and an associate editor of some peer-reviewed journals in computer science.