# Exploiting the Data Sensitivity of Neurometric Fidelity for Optimizing EEG Sensing

Zhen Ren, Xin Qi, Gang Zhou, *Senior Member, IEEE,* and Haining Wang, *Senior Member, IEEE*

***Abstract*—With newly developed wireless neuroheadsets, electroencephalography (EEG) neurometrics can be incorporated into *in situ* and ubiquitous physiological monitoring for human mental health. As a resource constraint system providing critical health services, the EEG headset design must consider both high application fidelity and energy efficiency. However, through empirical studies with an off-the-shelf Emotiv EPOC Neuroheadset, we uncover a mismatch between lossy EEG sensor communication and high neurometric application fidelity requirements. To tackle this problem, we study how to learn the sensitivity of neurometric application fidelity to EEG data. The learned sensitivity is used to develop two algorithms: 1) an energy minimization algorithm minimizing the energy usage in EEG sampling and networking while meeting applications' fidelity requirements and 2) a fidelity maximization algorithm maximizing the sum of all applications' fidelities through the incorporation and optimal utilization of a limited data buffer. The effectiveness of our proposed solutions is validated through trace-driven experiments.**

***Index Terms*—Data sensitivity analysis, electroencephalography (EEG) sensors, energy efficiency, neurometric fidelity.**

## NOMENCLATURE

| | |
|---|---|
| $a_i$ | $i$th application in the system, $i \in \{1 \dots N\}$. |
| $w_i$ | Weight associated with $a_i$. Larger weights are given to more important applications. |
| $s_j$ | $j$th data stream in the system, $j \in \{1 \dots M\}$. All $M$ data streams are assumed to have the same full sampling rate. |
| $S_i$ | Subset of data streams requested by $a_i$. Different applications may require different or the same set of data streams. |
| $t_j$ | Data decimation rate assigned to data stream $s_j$, $j \in \{1 \dots M\}$. |
| $T_i$ | $T_i = \{t_{i1}, t_{i2}, \dots\}$ denotes the data decimation assignment of the data streams set $S_i = \{s_{i1}, s_{i2}, \dots\}$, which is requested by $a_i$. |
| $f_i(T_i)$ | Fidelity function for $a_i$. The function input is the data decimation assignment $T_i$ to the data streams set $S_i$ that is requested by $a_i$. The function output is the $a_i$'s fidelity, which is in the range of [0,1]. |
| $F_i$ | Required fidelity threshold for $a_i$. |

Manuscript received December 09, 2013; revised April 02, 2014; accepted May 04, 2014. Date of publication May 6, 2014; date of current version June 04, 2014. This work was supported in part by the U.S. National Science Foundation under Grant CNS-1253506 (CAREER) and Grant CNS-1250180.

Z. Ren is with SynopSys Inc., Mountain View, CA 27703 USA (e-mail: zhen.ren@synopsys.com).

X. Qi, G. Zhou, and H. Wang are with the College of William and Mary, Williamsburg, VA 23187 USA (e-mail: xqi@cs.wm.edu; gzhou@cs.wm.edu; xqi@cs.wm.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JIOT.2014.2322331

## I. INTRODUCTION

THE electroencephalography (EEG) is able to record spontaneous neuro signals with multiple electrodes placed along the scalp, which provide insights into the understanding of human brain activities. In the clinical context, EEG has been widely used to monitor and diagnose various diseases, such as coma, stroke, and epileptic seizure. Psychological studies also indicate that depression and aging-related cognitive changes can be associated with EEG biomarks [1], [2]. With the mature of EEG sensing techniques, new opportunities have been emerging for developing effective monitoring and diagnosing applications for physiological and mental health. Especially, *in situ* physiological monitoring with EEG neurometrics is made possible by the recently developed low-cost wireless EEG sensor devices. Such an EEG device can be as small as a wireless headset or a headband, using battery as its power supply. It removes the need for cleaning the scalp area and hence requires minimum preparation for EEG collection. Without the electrode wires, it also frees patients from staying in the clinical environment, enabling ubiquitous and less invasive *in situ* physiological monitoring.

However, the new wireless EEG technology also brings new research challenges. First, since a wireless device no longer uses in-wall power supply but batteries, the energy usage of EEG data sampling and networking should be minimized. Second, even though the EEG data are transmitted through lossy wireless connections, almost all neurometric applications demand high application fidelity and tolerate little misdiagnosis. In comparison with other body sensor network applications such as daily activity recognition [3], a high volume of data traffic is usually collected through wireless in current EEG sensing applications [1], [2]. Therefore, it is essential to optimize the design of wireless EEG sensing for minimizing the impact of lossy wireless communication on application fidelity.

There are some existing works that design neurometric applications with a certain degree of fidelity. At the application level, algorithms for emotion detection and epileptic seizure detection are developed, and the application fidelity is evaluated in terms of detection accuracy [4], [5]. To balance sensing energy and application fidelities, a variety of approaches are proposed [6]–[9]. Some other works [10]–[13] trade application accuracy with energy efficiency through algorithm improvements. The tradeoff between energy consumption and data quality has also been studied in body sensor networks [14]–[16], but they only consider the distortion of sampled sensor readings rather than the application fidelity requirements. At a lower level, channel-aware QoS solutions have been developed for wireless sensor networks. For example, frequency adaptation [17] and encoding

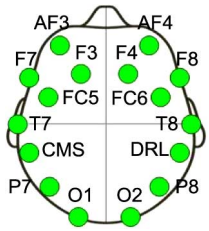



Fig. 1. Emotiv EPOC neuroheadset EEG electrodes positions.

adaptation [18] have been exploited for enhancing energy efficiency of communication. But, they do not associate the low level QoS metrics with the application fidelity requirements. In addition to communication solutions, other solutions are also proposed to improve energy efficiency by reducing the data transmission rate. For example, local processing is integrated in hardware design in [19] and compressive sensing is used in [20].

This paper aims to improve the communication design of EEG sensing applications while meeting applications' fidelity requirements. To this end, we investigate the sampling and networking of an off-the-shelf Emotiv EPOC Model 1.0 Neuroheadset [21] for neurometric applications.[1] We have studied several popular commercial wireless EEG headsets and headbands on the market, and have chosen to use the EPOC neuroheadset because of its high resolution and multiple EEG channels. As shown in Fig. 1, the neural signals sampled with 14 scalp sensors at the frequency of 128 Hz are most suitable for the medical applications, comparing with its peer products. The EPOC neuroheadset communicates directly with a nearby base station (a laptop is used in our experiments) plugged with a USB dongle. We measure the communication pattern between EEG sensors on the headset and the base station in real scenarios. Then, with the real traces we collected from the experiments, we investigate how the communication affects neurometric applications. We study two important neurometric applications: 1) the mild cognitive impairment (MCI) detection using Tsallis entropy (qEEG) ratio and 2) the depression detection using cerebral asymmetry score.

The MCI detection is based on the qEEG ratio. It is the ratio of prefrontal qEEG to occipital qEEG values, which are computed using averaged readings of electrodes in the two regions (AF3/AF4 in prefrontal cortex; O1/O2 in occipital lobe; see Fig. 1). According to [2], the qEEG ratio metric demonstrates statistically significant differences, and can be used to differentiate MCI patients from normal people. The depression detection is based on the cerebral asymmetry score, which quantifies the difference of the $\alpha$ band powers between the left and right cerebral hemispheres. Two symmetric electrodes F3/F4 are used to compute the asymmetry score. Studies in [1] propose an asymmetry score threshold for depression detection. Our evaluation shows that the current lossy communication has significant negative impact on the fidelity of these two neurometric applications.

Our design identifies three pitfalls that contribute to the mismatch between the lossy communication pattern and the neurometric application fidelity requirements. First, the current EEG wireless headset design does not consider the lossy communication patterns in reality, thus the application fidelity sharply drops when the wireless link quality degrades. Second, all electrodes of the EEG headset sample at the same speed without considering the sensitivity of applications' fidelity to different data streams. Like an I-frame is more significant than a B- or P-frame in an MPEG-4 video stream, different EEG sensing channels also have different impacts on the fidelity of different neurometric applications. But unlike the existing standardized I-/B-/P-frames in the MPEG-4 standard, there are no standardized data frames for different neurometric applications, and it is also difficult to standardize them. Therefore, a generic approach is needed to automatically learn the sensitivity of different neurometric applications' fidelities to the EEG sensory data. Third, applications' priorities are neglected. For instance, to a 29-year-old human subject with tentative diagnosis of depression, the electrode readings used for depression detection are more important than those used for the detection of aging-related cognitive changes.

To eliminate the above-mentioned pitfalls, we propose to quantify the sensitivity of neurometric application fidelity to the EEG data. The quantification results are used to improve the energy efficiency and neurometric fidelity of EEG sensor sampling and networking. Our main contributions are as follows.

1) We uncover and analyze a mismatch between the lossy EEG sensor communication pattern and the high neurometric application fidelity requirements.
2) More importantly, we propose to automatically learn the sensitivity of application fidelity to sampled sensor readings. While this learning technique itself can be generic for other applications, in this paper, we focus on neurometric applications.
3) With the learned sensitivity, we propose an energy minimization algorithm that allows us to minimize the energy usage in EEG sampling and MAC communication with given application fidelity requirements.
4) With the learned sensitivity, we also propose a fidelity maximization algorithm that allows us to maximize the sum of all applications' fidelities with a given data buffer on a wireless EEG headset.

This paper is organized as follows. Section II discusses related works and Section III presents our motivation. Section IV proposes to automatically learn the sensitivity of application fidelity to sensory data. Then, with the learned sensitivity, Section V minimizes energy consumption while meeting user-specified application fidelity requirements, and Section VI maximizes the sum of all applications' fidelities with a given data buffer on a wireless EEG headset. Section VII presents performance evaluation results, and Section VIII concludes this paper.

## II. Related Works

Several existing works propose neurometric applications with fidelity consideration. The emotion detection application developed in [4] aims to find the features that are robust to EEG signal noise and have strong discriminative capacity. In [5], an

[1]Note that the headset uses private hardware and software designs, and it does not provide direct communication control to developers. However, its manufacture encourages [21] researchers like us to give communication improvement suggestions which they will incorporate in future EEG sensing applications.

onset epileptic seizure detection algorithm is designed. It uses machine learning to extract spectral, spatial, and temporal features from sampled EEG signals to achieve high accuracy and short delay. However, to meet the application fidelity requirements, these works only consider improving the detection and classification algorithms, but do not consider lossy wireless communication. Moreover, at the presence of data loss, how to maximize application fidelity is not addressed.

At the application level, efforts have also been paid to trade accuracy with energy efficiency. In [10], a synchronization likelihood channel selection method is developed to reduce the number of EEG data streams used in emotion assessment, with only a slight loss of classification performance. In [11], a screening detector is developed to help multifeature detection algorithms reduce energy consumption by processing much fewer features. In [12], a machine learning technique is used to construct an epilepsy detector with fewer channels. In [13], lossless sensor data compression is proposed to reduce the communication energy usage at the expense of the increased computation complexity. However, different from our work, none of these solutions proposes to automatically learn the sensitivity of application fidelity to sensory data for minimizing energy consumption as well as maximizing the total application fidelity.

Sampling adaption methods that balance sensing energy and application fidelity have also been proposed for other human-centric applications. In [6], declarative rules are proposed to adapt sampling length of phone sensors. In [7], a context-aware speaker identification method uses a low sampling rate to detect whether a speaker exists. It switches to a high sampling rate for speaker identification when a speaker is detected. Similarly, SociableSense [8] lets the sensors operate at a high sampling rate only when interesting events happen. If there are no interesting events, the sensors are put to operate at a low sampling rate. More generally, AdaSense [9] combines lower power activity binary classification with higher power activity multi-classification to save sensing energy. In AdaSense, a sensor's sampling rate adapts with respect to different user activities. RadioSense [22] tunes packet sending rate to balance energy and accuracy for human activity recognition with wireless communication patterns. But, these works focus more on energy cost and do not consider communication issues.

The tradeoff between energy and data quality has been studied in body sensor networks. Barth *et al.* [14] evaluate the energy usage with respect to data distortion. Hanson *et al.* [15] explore energy-fidelity scalability to adjust compression ratios while maintaining data quality. Au *et al.* [16] provide real-time energy profiling and management for achieving desired sensing resolutions. However, these works quantify the fidelity with "mean square error" or "data resolution." These metrics only measure how much the raw sensor data are distorted but cannot quantify the distortion of the neurometric application fidelity, in terms of diagnosing accuracy and false alarm that we focus on.

At a lower level, channel-aware QoS solutions have been developed for enhancing the system performance. MMSN [17] provides frequency adaption to maximize parallel transmission among neighboring nodes for energy efficiency. Active collision recovery (ACR) [23] proposes adaptive encoding scheme to enhance collision recovery and transmission efficiency. The work in [24] boosts the wireless communication quality in body area network through jointly considering the packet sizes in both WiFi and ZigBee networks. However, these solutions do not associate the lower level QoS with application fidelity.

Fig. 2. Neuroheadset working with a laptop as base station.

Some other solutions are proposed to reduce the energy usage by reducing the data transmission rate. In [19], on-chip improvements have been proposed in EEG acquisition, digitization, and feature extraction. In [20], compressive sensing techniques are used to reduce the amount of sensing data to be transmitted. In [25] and [26], energy efficiency problems in wireless sensor network are also taken care of. These are not communication solutions like ours, but our proposed solution can be easily combined with these solutions to further optimize EEG sensing.

## III. Motivation

We use a newly developed commercial wireless EEG headset, the Emotiv EPOC Neuroheadset, to collect EEG signals. While the device is not specially designed for the usage of health care, it is the state-of-the-art high-resolution wireless neuro-signal acquisition device that we can get from the market. The neural signals are sampled by 14 scalp sensors, and transmitted using a custom wireless chipset operating in the 2.4-GHz band. A proprietary wireless connection is used to deliver the sampled data to a USB dongle, which then sends the data to a personal computer (PC) or a laptop base station through the USB port, as shown in Fig. 2. In this paper, we focus on the wireless communication link between the USB dongle and the EEG headset. According to the information from the Emotiv customer service, no packet retransmission scheme exists in its design. The headset's communication module does not open its control to developers, so we have conducted experiments in the home and office environments to observe its communication patterns.

In our experiments, we first study the real communication pattern of the headset. We observe that although the packet loss rate is low when the headset is connected to the base station, it can be easily disconnected. Then, we define the fidelities of two applications using two common neurometric indices, the Tsallis entropy (qEEG) ratio [2] and the cerebral asymmetry score [1]. Finally, based on our experiment results, we uncover a mismatch between the current wireless communication pattern and the applications' fidelity requirements.

### A. Communication Pattern

We collect 27 communication traces between the EEG headset and the local base station in both office and home environments. To ensure adequate environment diversity, we collect traces in different locations around offices and hallways in an academic

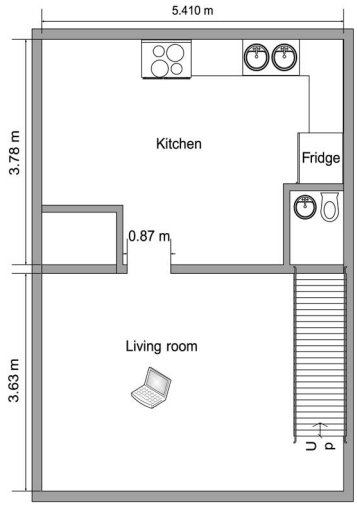



Fig. 3. Home environment for wireless neuroheadset communication trace collection.

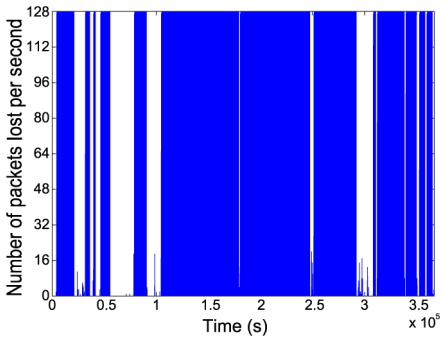



Fig. 4. Number of packet lost per second—communication trace of preparing food in kitchen.

building, and between the living room and the kitchen in a residential apartment. The communication traces are collected as the headset wearer performs various daily activities, such as walking, working, and cooking. From the traces, we have two observations.

*1) Easily Disconnected:* We observe that the communication between the EEG headset and the local base station is impacted by distance, human mobility, and environmental factors. The EEG headset's wireless communication range is very limited, and the headset is easily disconnected beyond 4–5 m. The trace in Fig. 4 is collected in home environment (see Fig. 3 for the apartment floor print) when the headset wearer was preparing food in the kitchen and moving around the house from time to time, with the local base station set in the middle of the living room. The $Y$-axis is the number of lost packets in each second. When the headset is disconnected, all 128 packets (the full sampling rate is 128 packets/s) are lost in each second during disconnection. In this trace, the headset is disconnected for about two-third of the time. The time elapsed before disconnection is short, ranging from 16 s to 21 min, with its cumulative distribution function (cdf) plotted in Fig. 5.

*2) Low Packet Loss Rate With Connection:* When the wireless communication is not disconnected, we divide the collected EEG data into bins of 1 s. Fig. 6 plots the cdf of the packet

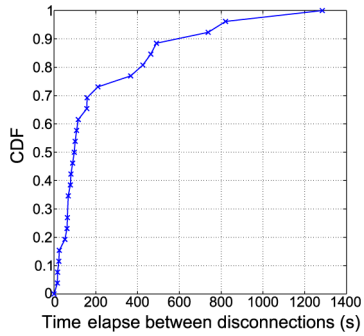



Fig. 5. CDF of connection length.

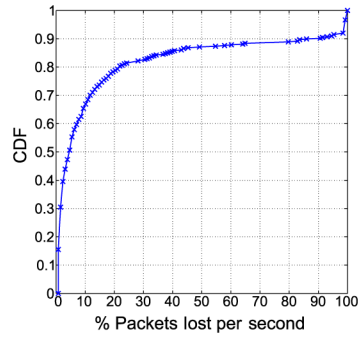



Fig. 6. CDF of packet loss rate for the Bins that contain packet loss (account for 93.90% of all Bins) during connection.

loss rate for the bins that contain packet losses, which only account for 6.05% of all used bins. From Fig. 6, we can see that about 80% of these bins have less than 20% packet lose. Considering that 93.95% of the bins do not even have any packet loss, only 1.18% of the total transmitted packets are lost. In other words, once the headset is connected, the packet loss rate is very low.

### *B. EEG-Based Neurometric Application and Fidelity Definition*

Based on two widely used neurometric indices, the Tsallis entropy ratio [2] and the cerebral asymmetry score [1], two neurometric applications are studied, and we give the definition for their application fidelities

$$\begin{aligned} \mathrm{var}_{\mathrm{Rapid}} &\equiv \sum_{x_i \in \mathrm{Interval}_j} \left(x_i - \bar{x}_j\right)^2 \\ \mathrm{var}_{\mathrm{Slow}} &\equiv \sum_{i=1}^{N} \left(x_i - \bar{x}\right)^2 \\ qEEG &= 1 - \frac{\sum_{\mathrm{Interval}_j} \mathrm{var}_{\mathrm{Rapid}}}{\mathrm{var}_{\mathrm{Slow}}} \\ R_{\mathrm{po}} &= \frac{\text{prefrontal qEEG}}{\text{occipital qEEG}}. \end{aligned} \tag{1}$$

*1) MCI Detection Using Tsallis Entropy (qEEG) Ratio:* Equation (1) shows how to compute qEEG ration metric. First, we compute the qEEG from prefrontal and occipital channels, respectively. The qEEG characterizes the EEG signal variance in slow and rapid manners. We divide time

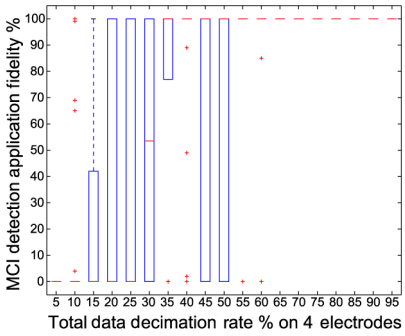



Fig. 7. MCI detection accuracy change with different data decimation rate.

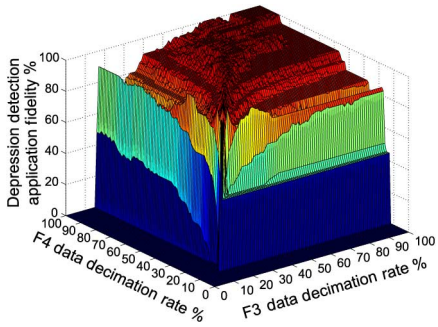



Fig. 8. Depression detection accuracy change with different EEG data sampling rate.

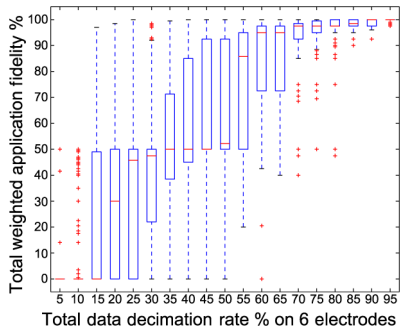



Fig. 9. Total detection accuracy change with different EEG data sampling rate.

into epochs of 30 s, the same as used in [2], and the local maxima and minima of the averaged readings further divide each epoch into intervals. We compute qEEG using the variance values in each epoch and interval. Then, the qEEG ratio equals the ratio of prefrontal qEEG to occipital qEEG. Finally, according to [2], statistical hypothesis tests (two tailed T-tests) are used to decide if the mean of the subject's $R_{\rm po}$ is below or above the statistic threshold $\mu_0$ in [2].

Now, we define and quantify the application fidelity of MCI detection. We define the application fidelity as the MCI detection accuracy. To quantify the application fidelity, we use the result of classification with full information, i.e., EEG signals sampled with the full rate, as the ground truth. Then, using partial information, i.e., EEG signals with reduced sampling rate, we perform the MCI classification again. Here, we use the term *Data Decimation* to refer to retaining a lower rate than the full rate for data sampling and transmission. The data decimation rate is computed as the ratio of the new sampling rate after decimation compared with the full rate. For example, 10% data decimation means that the data sampling and transmission rate is reduced to 10% of the full rate. With the full-sampling-rate traces we collected, the data decimation is performed by randomly keeping a portion of the EEG readings and dropping the rest. Finally, we calculate the percentage of the classification results that are identical to the ground truth. This percentage is defined to be the application fidelity of MCI detection.

Fig. 7 presents the fidelity of MCI detection. Since for a given total data decimation rate, different combinations of data decimation rates (which we call data decimation assignments) can be applied to the same four electrodes, we randomly choose 30 of the possible data decimation assignments and plot their fidelity results with box plot in Fig. 7. This box plot shows the median, 25th and 75th percentiles, upper and lower adjacent values, and outliers. As the figure shows, the application fidelity is low when the data decimation rate is below 10%, and changes dramatically with the low data decimation rate between 15% and 30%. This implies that the samples delivered to the base station are not sufficient for the MCI detection. Even with higher data decimation rate, the decimation rate of each channel may not be the same. Given some channels have too low data decimation rates and the samplings are insufficient, the application fidelity still suffers even if other channels have data decimation rates higher than necessary, showing the large variation of application fidelity even with relatively high total data decimation rates, such as 45%–50%.

*2) Depression Detection Using Cerebral Asymmetry Score:* The cerebral asymmetry score is computed with the readings from F3/F4. We divide time into chunks of 2.05 s, with 75% overlapping [1], and apply fast Fourier transform [27] to the electrode readings in each chunk. The $\alpha$ band power (measured in $\mu V^2/Hz$) is computed by summing up activities across all frequencies between 8 and 13 Hz. Log-transformation is applied for normalization. The asymmetry score equals the log of the right hemisphere $\alpha$ band power minus the log of the left hemisphere $\alpha$ band power (log of the right $\alpha$ band power $-$log of the left $\alpha$ band power in $\mu V^2/Hz$). Using $T$-test, the mean of asymmetry score is compared with the thresholds in [1] to generate depression detection results.

Similar to the MCI detection, the fidelity of the depression detection application is defined as the percentage of identical event detection results between when the full-sampling-rate data are used and when decimated data are used. Fig. 8 plots the fidelity of asymmetry score when different data decimation rates are applied.

Assuming both applications are equally important, each with 0.5 priority weight, Fig. 9 plots the total weighted fidelity with data decimation rates on all six electrodes, using box plot.

### C. Mismatch Between the Communication Pattern and Application Fidelity

The communication pattern analysis shows that the wireless EEG headset is either well connected or disconnected with the local base station most of the time. So, we discuss the application fidelity under these two scenarios.

*1) Well Connected:* We find that the current design uses a higher sampling rate than necessary. As we apply data decimation

to the sampled data, the result shows that it is still possible to maintain high fidelity for the two example applications. For example, in Fig. 8, the fidelity of depression detection can still be as high as 94% when both F3 and F4 electrodes' readings are applied with 15% data decimation rate. This implies that the application does not need the full sampling rate of 128 Hz for all the electrodes on the neuroheadset. Figs. 7–9 show that the application fidelity is generally monotonous with the data decimation rate, but can be degraded with inappropriate data decimation assignments. This means that different data streams have different impacts on specific application's fidelity. For instance, when F4 electrode maintains the full sampling rate and F3 only applies 20% data decimation rate, the depression detection fidelity drops sharply to 39%. The fidelity is much worse than what is given in our previous example, although the total data decimation rate of the two electrodes is much higher (a total data decimation rate of 40% comparing with 15%). Fig. 7 also shows that even when the total data decimation rate is fixed, the fidelity may still vary with different data decimation assignments, e.g., with the total data decimation rate fixed as 45%, the application fidelity can vary from zero to 100%.

*2) Disconnected:* We find that the current design overlooks application fidelity requirements when the headset is disconnected. The Emotiv neuroheadset does not have any data buffer. When the headset is disconnected, the sampled data are completely lost, and the application fidelity is not provided during the disconnection period. As a result, the use of EEG headset will be interrupted when the subject cannot remain close enough to the local base station. For example, in the kitchen cooking trace we collected, the subject needs to wear the headset for more than 45 min to perform a 15-min effective EEG monitoring.

*3) Summary:* Based on the measurement results and analysis, we can see that the existing commercial neuroheadset does not take the realistic communication patterns and application fidelity requirements into its design consideration. The current neural signal sampling design is inadequate for meeting the application fidelity requirements and handling real-time communication scenarios. Thus, we propose a new approach, which can automatically learn the sensitivity of application fidelity to EEG data, and utilize the learned sensitivity to cope with the mismatch between the neuroheadset's wireless communication pattern and the application fidelity requirements. Note that our approach is not limited to the Emotiv EPOC Neuroheadset, as it does not depend on any specific design of the wireless EEG device.

## IV. Learning the Sensitivity of Single Application's Fidelity

With the two example applications, we further analyze the sensitivity of specific application's fidelity to the EEG data. According to the preliminary observations in Section III, different data streams have different impacts on the application fidelity. In other words, an application's fidelity may be more sensitive to the data decimation rate changes of some streams than others. So, we propose to learn the sensitivity of application fidelity, and also use it to optimize resource allocation in EEG sensing. Nomenclature lists the notations used in this section.

First, we look into individual applications using multiple data streams. If the application fidelity is more sensitive to some data streams, then the sampling rates of these streams should be reduced as less as possible. Given a total data decimation rate, we formally define the problem of computing the optimal data decimation assignment for a single application as follows.

*Definition 1 (Single-Application Decimation)*: Given the total data decimation rate $l_i = \frac{1}{|S_i|}\sum_{\forall t_j \in T_i} t_j$, the problem is to find the optimal data decimation assignment $T_i = \{t_{i1}, t_{i2}, \ldots\}$ for $a_i$, so that its fidelity $f_i$ is maximized.

We notice that this problem is similar to the traditional sensitivity analysis [28], [29], which tries to identify the relative importance of each input (corresponding to each data stream in our problem). So, a basic sensitivity analysis technique called Local Method is used here to assign the data decimation rate to different data streams of a single application. To decide the importance of each data stream $s_j$, the Local Method checks the simple derivative of the fidelity function $f_i$ with respect to $s_j$'s data decimation rate change. While all other data streams use the same sampling rate, $s_j$ is applied with data decimation rate $t_j$ which is changed in $K$ enumeration steps. We denote the change between the $k$th and $(k+1)$th data decimation rates as $\Delta t_{jk}$, and the corresponding application fidelity change as $\Delta f_{ik}$, $k \in \{1, \ldots, K-1\}$. Then, the average derivative for data stream $s_j$ is $\frac{1}{K-1}\sum_{k=1}^{K-1} |\frac{\Delta f_{ik}}{\Delta t_{jk}}|$. Intuitively, with a larger value of the average derivative, the data stream $s_j$ is more important to the application. Thus, the data decimation rate of data streams $j$ is computed as

$$t_j = \frac{l_i \times |S_i| \times \sum_{k=1}^{K-1} \left|\frac{\Delta f_{ik}}{\Delta t_{jk}}\right|}{\sum_{\forall t_j \in T_i} \sum_{k=1}^{K-1} \left|\frac{\Delta f_{ik}}{\Delta t_{jk}}\right|}. \quad (2)$$

With the learned data sensitivity, we further consider multiple applications in the EEG sensing system. Depending on whether the wireless link is well connected or disconnected, we address two optimization problems. 1) As we have illustrated in Section III, full data sampling rate is usually not needed to produce the requested application fidelity. So, we propose to optimize the total data decimation rate of all data streams while meeting the user requested application fidelity. Obviously, minimizing the total data decimation rate lowers the total energy usage, which is essential for the battery powered wireless EEG headset. 2) In the cases, when the wireless communication is disconnected, a local buffer is incorporated to temporarily cache the data for later delivery. Since the buffer space is limited, here we propose to optimize the usage of the limited space by optimizing the data decimation assignment among different data streams, so that the weighted sum of all applications' fidelities is maximized.

## V. Exploiting Data Sensitivity to Minimize Energy Consumption

Since data communication and processing dominate the energy cost of a sensing system, we assume that the total energy

consumption is approximately proportional to the amount of data the sensing system needs to communicate and process. In this section, we define the energy minimization problem when the wireless communication link is reliable, and propose an energy minimization algorithm (mainly for the MAC layer) by minimizing the total data decimation rate.

### A. Problem Definition and Analysis

When the wireless link is well connected, we formally define the energy minimization problem as follows.

*Definition 2 (Energy Minimization Problem)*: For the purpose of minimizing energy consumption, how to optimize the data decimation assignment $T_i$ to the data streams used by application $a_i$, so that each $a_i$'s desired fidelity threshold $F_i$ is satisfied and the total decimation rate $\frac{1}{M}\sum_{j=1}^{M} t_j$ is minimized.

Note that a higher data decimation rate (or more data) implies more communication and energy cost. Moreover, this problem can be formalized as follows:

$$\min\left\{\frac{1}{M}\sum_{j=1}^{M} t_j\right\}$$
$$\text{s.t.} \quad \forall i,\ f_i(T_i) \geq F_i.$$

In this case, the constraint functions are described by *oracle models*.

In an oracle model, we do not know the $f_i$ functions explicitly, but we can evaluate $f_i$ functions manually or by computer programs. This is referred to as querying the oracle [30]. In [30], it is also mentioned that some prior information such as an oracle model's convexity is sometimes given or could be assumed depending on the application context. Generally, Fig. 8 shows that a higher data decimation rate (more communication energy) means higher application fidelity. Thus, it is reasonable for us to assume that the fidelity function is an approximately increasing function of the assigned data decimation rates.

### B. Energy Minimization Algorithm

We propose Algorithm 1 to solve the energy minimization problem. Considering all applications in the system, we propose to adjust the data decimation rates across data streams used by different applications to minimize the total data decimation rate while still meeting the desired application fidelity thresholds. This algorithm uses the Local Method in Section IV to learn the sensitivity of an application's fidelity and determine the optimal data decimation assignment for the data streams required by this application.

From Figs. 7 and 8, we find that the application fidelity generally increases with the increase of data decimation rates. Based on this monotonous property, we design the main algorithm for multiapplication data decimation assignment, which is inspired by the bisection method for quasiconvex optimization [30]. This algorithm utilizes an approach similar to the binary search, which repeatedly adjusts the data decimation rate $l_i$ for all data streams requested by each application $a_i$. This loop stops when all the resulting applications' fidelities $\{f_i\}$s are above and close enough to the required thresholds $\{F_i\}$s within a specified tolerance interval $\epsilon(\epsilon > 0)$. With each possible $l_i$, the data decimation rate is assigned to the data streams of $a_i$ according to (2).

The convergence speed of the algorithm depends on the datasets it runs on. For the datasets we use in the evaluation, the algorithm does converge after running for several minutes. Additionally, we can also set a termination condition, such as the number of total iterations, into the algorithm to control its complexity.

**Algorithm 1** Energy Minimization

**Input:** fidelity functions $\{f_i(T_i)\}$, fidelity thresholds $\{F_i\}$, the tolerance $\epsilon$

**Output:** data decimation assignment to all data streams $\{t_1..t_M\}$

$\forall j = 1..M, t_j = 0$

**for** $i = 1$ to $N$ **do**

  Initiate the decimation rate upper / lower bounds: $u_i = 1; b_i = 0$

  Compute the initial decimation rate for $a_i$: $l_i = \frac{b_i+u_i}{2}$;

  **repeat**

    /*Assign the total data decimation rate $l_i$ of application $a_i$ to all data streams that application $a_i$ requests, according to Equation 2*/

    **for** $t_j \in T_i$ **do**

      **if** $t_j <$ the result of Equation 2 **then**

        $t_j =$ the result of Equation 2

      **end if**

    **end for**

    Compute $f_i(T_i)$

    **if** $f_i - F_i > 0$ **then**

      **if** $f_i - F_i < \epsilon$ **then**

        Break

      **end if**

      $u_i = l_i$

    **else if** $f_i - F_i < 0$ **then**

      $b_i = l_i$

    **else**

      Break

    **end if**

    $l_i = \frac{b_i+u_i}{2}$

  **until** $false$

**end for**

Return (the total data decimation rate $T = \frac{1}{M}\sum_{j=1}^{M} t_j$)

## VI. Exploiting Data Sensitivity to Maximize Application Fidelity

When the data with the full sampling rate cannot be all immediately delivered due to network disconnections, the MAC layer uses a local buffer to cache data for later delivery when the network is reconnected. In this section, we define the application fidelity maximization problem, and propose a novel algorithm to optimize the usage of the limited buffer to maximize the total weighted application fidelity.

### A. Problem Definition and Analysis

When the wireless communication is disconnected, we define the application fidelity maximization problem:

*Definition 3 (Fidelity Maximization Problem)*: Given a data buffer size, which determines the upper threshold for $\frac{1}{M}\sum_{j=1}^{M} t_j$, how to optimize the data decimation assignment $\{t_1, \ldots, t_M\}$ to data streams, so that the sum of all applications' weighted fidelity $\sum_{i=1}^{N} w_i \cdot f_i(T_i)$ is maximized.

Based on the definition, we find that it is a variant of the resource allocation problem (RAP). RAP aims to assign available resources to all agents in an economic way. Many algorithms exist for solving RAP. One class of resource allocation algorithms among them is the weighted majority [31] whose basic skeleton is described in Algorithm 2.

**Algorithm 2** Weighted Majority Algorithm Skeleton

1: **Input:** $\beta \in [0,1]$, the initial weight vector $\mathbf{u^1} \in [0,1]^N$ with $\sum_{i=1}^{N} u_i^1 = 1$, the number of trials $T$
2: **Output:** the allocation vector $\mathbf{p^t}$
3: **for** $t = 1$ to T **do**
4: Choose allocation $\mathbf{p^t} = \frac{\mathbf{u^t}}{\sum_{i=1}^{N} u_i^t}$
5: Receive the loss vector $\mathbf{l^t} \in [0,1]^N$ from environment
6: Set the new weights vector $u_i^{t+1} = u_i^t \beta^{l_i^t}$
7: **end for**

In Algorithm 2, the weight vector represents how many resources should be allocated to each agent (application), and all resources are initially evenly allocated. After initialization, the algorithm goes into the allocation loop. In each round of the loop, losses from all agents are put to form a loss vector. With the loss vector, the algorithm adjusts each agent's weight according to the following rule: allocating more resources to the agent who makes profits, and reducing resources to those agents who do not. The parameter $\beta$ is used to control how fast the algorithm converges. When the next round of the loop begins, the resources are reallocated according to the new weight vector. The algorithm terminates after a certain number of rounds and the final weight vector represents the final decision on how to allocate resources to the agents.

Since the weighted majority algorithm has promising and solid convergence analysis result, we design an algorithm based on it to solve the fidelity maximization problem.

**Algorithm 3** Fidelity Maximization

**Input:** fidelity functions $\{f_i(T_i)\}$, the total decimation rate $T$, the iteration bound $Q$, $\beta \in [0,1]$ ($\beta$ controls the algorithm's converging speed)
**Output:** data decimation assignment to all data streams $\{t_1..t_M\}$
/* $\forall i$, initiate $\{f_i^0\}$, $\{l_i\}$, and the fidelity change $\Delta f_i$*/
$\forall i$, $f_i^0 = 1$, $l_i = T$, $\Delta f_i = 0$
/* Initially, assume $\forall j, t_j = T$ */
$\forall j$, $t_j^1 = T$
Initiate the iteration step index $q = 1$
**repeat**
  $\forall i$, compute $f_i^q$ with $\{t_j^q\}$
  /* If the total weighted fidelity decreases, recompute $\Delta f_i$*/
  **if** $\sum_{i=1}^{N} f_i^{q-1} \cdot w_i - \sum_{i=1}^{N} f_i^q \cdot w_i > 0$ **then**
    $\forall i$, compute $\Delta f_i = (f_i^{q-1} - f_i^q) \cdot w_i$ for application $a_i$
  **end if**
  /* Adjust $l_i$ for each application $a_i$, according to $\Delta f_i$*/
  **for** $i = 1$ to $N$ **do**
    **if** $\Delta f_i > 0$ **then**
      $l_i = l_i * \beta^{-\Delta f_i}$
    **end if**
    /* Compute the $q^{\text{th}}$ data decimation assignment for application $a_i$ with the adjusted $l_i$, according to Equation 2 */
    **for** $j = i1, i2, ..$ **do**
      **if** $t_j^q$ is NULL OR $t_j^q <$ the result of Equation 2 **then**
        $t_j^q =$ the result of Equation 2
      **end if**
    **end for**
  **end for**
  /* Normalize $\{t_j^q\}$*/
  $\forall j, t_j^q = \frac{T \cdot M \cdot t_j^q}{\sum_{j=1}^{M} t_j^q}$
  $q = q + 1$
**until** $q > Q$
Return (the data decimation assignment $\{t_1, .., t_M\}$)

### B. Fidelity Maximization Algorithm

We propose Algorithm 3 to solve the fidelity maximization problem. Given the total possible data decimation rate $T$, which is decided by the available buffer size, our algorithm iteratively adjusts the data decimation assignment among applications.

From each iteration, the trends of changes are learned for both the total weighted fidelity and individual applications' fidelities. Then, the algorithm increases the data decimation rate of those applications that negatively impact the total weighted fidelity. It also follows (2) to compute the data decimation assignment to data streams requested by each application.

As the algorithm iteratively computes the data decimation assignment $\{t_1, .., t_M\}$, we denote the index of the current iteration step as $q$. The results of the $q$th iteration are denoted with superscript $q$. The complexities of Algorithms 1 and 3 rely on the convergence speed, which depends on the datasets they run on. Thus, we have configurations such as the iteration bound Q in Algorithm 3 in order to control the algorithm complexities. The algorithm terminates iteration after a specified number of rounds $Q$. So, $q \in \{1, \ldots, Q\}$.

Like many other algorithms for solving complicated optimization problems, our proposed algorithm may end at a local minimum. We could design an algorithm that can achieve a global minimum when we know whether the problem is convex or not. However, as we mentioned before, we do not have explicitly mathematical equations for describing the optimization objectives. Thus, it is hard for us to determine the convexity of the problem. Therefore, we evaluate the functions in the optimization objectives for solving the problems numerically [30].

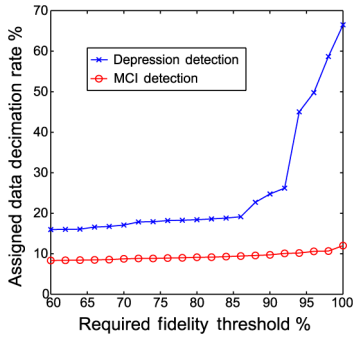



Fig. 10. Data decimation rates computed with energy minimization algorithm.

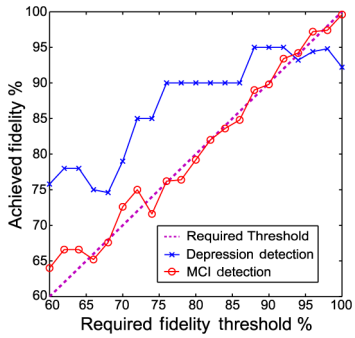



Fig. 11. Fidelity achieved with energy minimization algorithm.

## VII. Evaluation

We evaluate the above two proposed solutions with trace-driven experiments, using 20 EEG data traces collected from a human subject in the same office environment, where we collected the traces for the communication patten analysis in Section III. Each trace contains nonloss EEG data generated by six EEG electrodes. We apply data decimation to the traces with the data decimation assignments computed from our two algorithms. Then the application fidelity is calculated with decimated traces, and compared with the current solutions that are implemented in the EEG headset, i.e., the default full data rate sampling and even data decimation assignments.

Algorithms 1 and 3 are evaluated using the 20 traces, which are separated into a training set and an evaluation set. Each algorithm first runs on the training set to compute the data decimation assignment. Then, the computed data decimation assignment is applied to the evaluation set. On both sets, the fidelity is computed as defined in Section III. The evaluation is repeated following a fivefold cross-validation style, i.e., the 20 are divided into five groups with four traces in each group; each of the five groups is used in turn as the evaluation set, and the other four groups are used as the training set so that the data for evaluating the algorithms is different from the data for training the algorithms. In total, each experiment runs five rounds, and the fidelities achieved in these five rounds are averaged.

### A. Evaluation of the Energy Minimization Algorithm

Figs. 10 and 11 show the results of the Algorithm 1, which achieves large energy savings comparing with the original design of using the full sampling rate for all electrodes at all times. The $x$-axis of the two figures shows that the user-specified application fidelity requirement increases from 60% to 100%. Fig. 10 shows the total data decimation rate for all electrodes (two for depression detection; four for MCI detection) used by each application with required fidelity thresholds. Fig. 11 shows the achieved application fidelity when the data decimation assignment computed by Algorithm 1 is applied to the evaluation set. We observe that both applications approximately achieve the required neurometric fidelities with the computed data decimation assignment, which is lower than the full sampling rate. So, energy can be saved by sampling and sending the decimated data only. We observe that when the required threshold is around 90%, only less than 25% and 10% data decimation rates are needed for the two applications, respectively. This means, 75% and 90% of the sampling and communication energy can be saved by our solution for each application.

Fig. 10 also shows the effectiveness of utilizing different sensitivities of different applications' fidelities to EEG data. The computed data decimation rate increases as the required fidelity threshold increases, and the data decimation rate for depression detection needs to be largely increased to achieve a fidelity higher than 85%. This implies that the fidelity of depression detection is more sensitive to EEG data than the MCI detection. So, when a higher application fidelity is required, a higher data decimation rate should be assigned to the data that is requested by depression detection.

Fig. 11 shows the fidelity achieved by the data decimation assignment computed by Algorithm 1. Most of the achieved

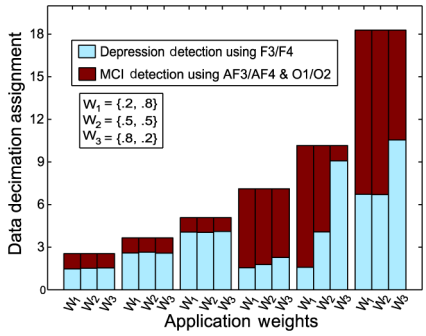



Fig. 12. Data decimation assignment computed with fidelity maximization algorithm.

fidelity is higher than or close to the required threshold for both applications. For MCI detection, the achieved fidelity is always slightly higher than or roughly equal to the required fidelity. For depression detection, the achieved fidelity is higher than the required fidelity when the requested threshold is less than 92%, after which the requested high threshold is not met. One possible reason is that the learned sensitivity of fidelity is only an approximate, but not a strict, increasing function of the data decimation rates (see Fig. 8).

### *B. Evaluation of the Fidelity Maximization Algorithm*

We compare our fidelity maximization algorithm with the default even data decimation assignment algorithm, which assigns the total data decimation rate evenly to the six needed electrodes. When the wireless communication is reliable, we use the energy minimization algorithm to achieve perfect application fidelity with the minimum energy consumption. When the wireless communication is disconnected, we incorporate a limited data buffer into the EEG headset to save sampled data temporarily for later delivery. During the disconnection period, the fidelity maximization algorithm is also called to maximize the total weighted application fidelity according to the limited buffer size. As demonstrated in Fig. 10, only a small data decimation rate is needed in order to achieve the perfect application fidelity. This suggests that we need a smart way to allow us to focus on a small range of low data decimation rate in order to effectively evaluate our fidelity maximization algorithm. Therefore, in this experiment, we first apply our energy minimization algorithm to get the minimally needed data decimation rate. Then, we further apply a second round of decimation with given data decimation rates. The given data decimation rate is decided by the fixed buffer size and the length of the disconnection period. As Fig. 4 shows, the disconnection periods varies. The longer the disconnection period, the lower the data decimation rate is given. We present the results for six test cases with six different data decimation rates given, including 12.5%, 18%, 25%, 35%, 50%, and 90%, which correspond to the different disconnection periods, respectively. Different application weights are also allowed for the two applications. For example, a weight vector $W_1 = \{0.2, 0.8\}$ means that the depression detection application and the MCI detection application have priority weights of 0.2 and 0.8, respectively. Fig. 12 plots the computed data decimation assignments and Fig. 13 plots the achieved application fidelity.

In Fig. 12, the three bars in each of the six test cases show the data decimation assignments computed with one of the six given total data decimation rates as well as the three different combinations of the application weights. The figure shows that the fidelity maximization algorithm is able to adapt the data decimation assignment to meet different application fidelity and data decimation rate requirements. In the first three test cases, we find that similar data decimation assignments are computed for the MCI detection application. This suggests that with a very low total data decimation rate ($\leq 25\%$), the application's fidelity cannot be largely improved with a small data decimation rate increase. So, the algorithm assigns higher data decimation rates to the application when it becomes more important. This is shown in the first bar of each test case that corresponds to the weight vector $W_1 = \{0.2, 0.8\}$. When the depression detection application becomes more important, as shown in the third bar of each test case that corresponds to the weight vector $W_3 = \{0.8, 0.2\}$, the algorithm assigns higher data decimation rates to the depression detection application.

From Fig. 13, when the buffer size is very small, as shown in the first three test cases of each subfigure, we observe that the default even data decimation assignment algorithm achieves a very low fidelity and is always largely outperformed by our proposed algorithm. This is because, as demonstrated in Fig. 10, the MCI detection application can easily achieve a very high fidelity with a very low data decimation rate compared with the depression detection application. Our proposed algorithm is able to automatically learn the sensitivities of different applications' fidelities to the EEG data. Then, based on the learned sensitivity, our algorithm chooses to first satisfy the application that requires lower data decimation rates to achieve a high application fidelity. On the other hand, the default algorithm assigns the same data decimation rates to all data streams, ignoring the sensitivity difference between different applications' fidelities. Thus, data decimation rates higher than necessary are assigned to the data streams of one application, but the other application's fidelity is harmed because of inadequate data decimation rates assigned.

From Fig. 13, when the buffer size is not very small, as shown in the last three test cases of each subfigure, we observe that our design is still better than the default solution most of the time. In these three test cases, both algorithms assign high data decimation rates to both applications because the total allowed data decimation rate is high. With the high data decimation rates assigned, the MCI detection application's fidelities achieved by both solutions are close to 100%, and the depression detection's fidelities achieved by these two solutions do not have obvious difference. This is because, as shown in Fig. 10, when the data decimation rate is higher than or equal to 20%, a small increase of the data decimation rate does not lead to a large improvement of the depression detection application's fidelity.

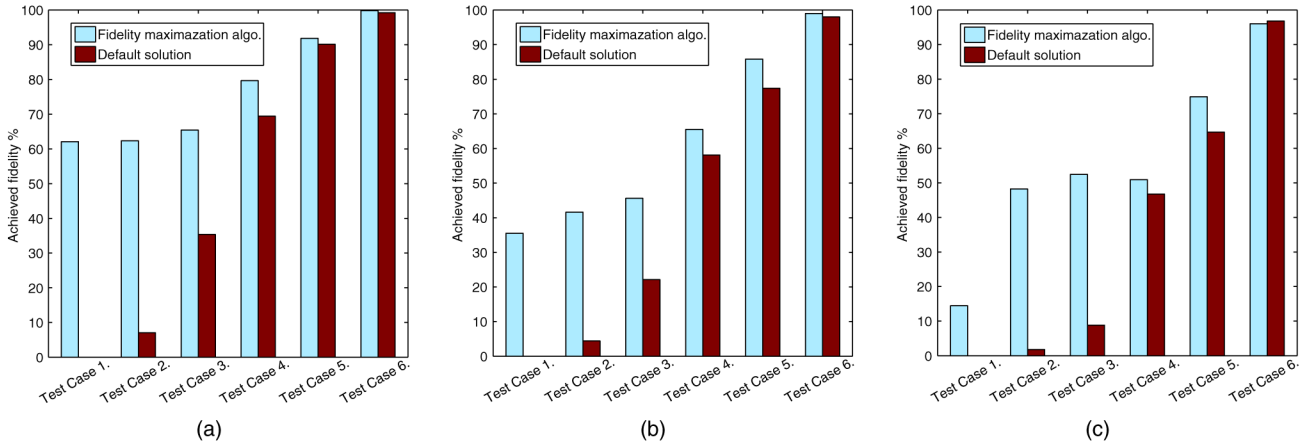



Fig. 13. Fidelity achieved with fidelity maximization algorithm. (a) Application weight $W_1 = \{0.2, 0.8\}$. (b) Application weight $W_2 = \{0.5, 0.5\}$. (c) Application weight $W_3 = \{0.8, 0.2\}$.

## VIII. Conclusion and Future Work

Meeting applications' fidelity requirements and saving energy are two central issues of incorporating EEG neurometrics into *in situ* and ubiquitous physiological monitoring. In this paper, we measure the realistic neuroheadset communication with an off-the-shelf Emotiv EPOC Neuroheadset, and reveal a mismatch between the lossy EEG sensor communication pattern and the high neurometric application fidelity requirements. Then, taking the MCI detection and the depression detection as two example neurometric applications, we propose a generic approach to automatically learn the sensitivity of application fidelity to the EEG data. With the learned sensitivity, we propose an energy minimization algorithm for reliable wireless communication. We also propose a fidelity maximization algorithm for poor wireless communication. Through trace-driven experiments, our solutions are demonstrated to outperform existing ones. When the communication is reliable, the energy minimization algorithm is used to save energy. When the communication is poor, the fidelity maximization algorithm is used to maximize application fidelity. In the future work, we will investigate more on the joint use of the two algorithms. We will devise a new communication indicator, and use it to guide dynamic switches between these two algorithms. Also, we will further analyze the performance of our algorithms. Since both algorithms' convergence speeds depend heavily on the input EEG data, we will evaluate our solutions with a larger set of EEG data from multiple subjects to figure out the proper configurations, such as the iteration bound Q, for our proposed algorithms.

## References

[1] J. Henriques and R. J. Davidson, "Left frontal hypoactivation in depression," *J. Abnormal Psychol.*, vol. 100, no. 4, pp. 535–545, 1991.

[2] T. J. D. Bock *et al.*, "Early detection of Alzheimer's disease using nonlinear analysis of EEG via Tsallis entropy," in *Proc. IEEE Biomed. Sci. Eng. Conf. (BSEC)*, 2010, pp. 1–4.

[3] M. Keally, G. Zhou, G. Xing, J. Wu, and A. Pyles, "PBN: Towards practical activity recognition using smartphone-based body sensor networks," in *Proc. 9th ACM Conf. Embedded Netw. Sensor Syst. (ACM SenSys)*, 2011, pp. 246–259.

[4] M. Mikhail *et al.*, "Emotion detection using noisy EEG data," in *Proc. 1st Augmented Human Int. Conf. (ACM AH)*, 2010, pp. 7:1–7:7.

[5] A. Shoeb and J. Guttag, "Application of machine learning to epileptic seizure detection," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 975–982.

[6] K. K. Rachuri *et al.*, "Emotionsense: A mobile phones based adaptive platform for experimental social psychology research," in *Proc. ACM Ubicomp*, 2010, pp. 281–290.

[7] H. Lu *et al.*, "Speakersense: Energy efficient unobtrusive speaker identification on mobile phones," in *Proc. 9th Int. Conf. Pervasive Comput.*, 2011, pp. 188–205.

[8] K. K. Rachuri *et al.*, "Sociablesense: Exploring the trade-offs of adaptive sampling and computation offloading for social sensing," in *Proc. ACM MobiCom*, 2011, pp. 73–84.

[9] X. Qi, M. Keally, G. Zhou, Y. Li, and Z. Ren, "Adasense: Adapting sampling rates for activity recognition in body sensor networks." in *Proc. 19th IEEE Real-Time Embedded Technol. Appl. Symp.*, 2013.

[10] K. Ansari-Asl, G. Chanel, and T. Pun, "A channel selection method for EEG classification in emotion assessment based on synchronization likelihood," in *Proc. 15th Eur. Signal Process. Conf. (EUSIPCO)*, 2007, pp. 1241–1245.

[11] E. I. Shih and J. J. Guttag, "Reducing energy consumption of multi-channel mobile medical monitoring algorithms," in *Proc. 2nd Int. Workshop Syst. Netw. Support Health Care Assisted Living Environ. (ACM HealthNet)*, 2008, pp. 15:1–15:7.

[12] E. I. Shih, A. H. Shoeb, and J. V. Guttag, "Sensor selection for energy-efficient ambulatory medical monitoring," in *Proc. 7th Int. Conf. Mobile Syst. Appl. Serv. (ACM MobiSys)*, 2009, pp. 347–358.

[13] K. C. Barr and K. Asanović, "Energy-aware lossless data compression," in *Proc. ACM TOCS*, 2006, pp. 250–291.

[14] A. T. Barth, M. A. Hanson, H. C. P. Jr, and J. Lach, "Online data and execution profiling for dynamic energy-fidelity optimization in body sensor networks," in *Proc. IEEE Int. Conf. Body Sensor Netw. (BSN)*, 2010, pp. 213–218.

[15] M. Hanson, H. Powell, A. Barth, and J. Lach, "Enabling data-centric energy-fidelity scalability in wireless body area sensor networks," in *Proc. 4th Int. Conf. Body Area Netw. (BodyNets)*, 2009, pp. 16:1–16:8.

[16] L. K. Au *et al.*, "Microleap: Energy-aware wireless sensor platform for biomedical sensing applications," in *Proc. IEEE Biomed. Circuits Syst. Conf. (BIOCAS)*, 2007, pp. 158–162.

[17] G. Zhou *et al.*, "MMSN: Multi-frequency media access control for wireless sensor networks," in *Proc. IEEE 25th Int. Conf. Comput. Commun. (INFOCOM)*, 2006, pp. 1–13.

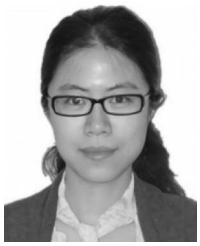
[18] C.-J. Liang, N. Priyantha, J. Liu, and A. Terzis, "Surviving Wi-Fi interference in low power ZigBee networks," in *Proc. 8th ACM Conf. Embedded Netw. Sensor Syst. (SenSys)*, 2010, pp. 309–322.

[19] N. Verma, A. Shoeb, J. V. Guttag, and A. P. Chandrakasan, "A micro-power EEG Acquisition SoC with integrated seizure detection processor for continuous patient monitoring," in *Proc. IEEE Symp. VLSI Circuits*, 2009, pp. 62–63.

[20] Z. Charbiwala, V. Karkare, S. Gibson, D. Markovic, and M. B. Srivastava, "Compressive sensing of neural action potentials using a learned union of supports," in *Proc. IEEE Int. Conf. Body Sensor Netw. (BSN)*, 2011, pp. 53–58.

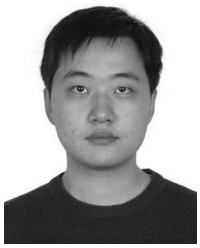
[21] (2012). *Emotiv EPOC Neuroheadset* [Online]. Available: http://www.emotiv.com/

[22] X. Qi, G. Zhou, Y. Li, and G. Peng, "Radiosense: Exploiting wireless communication patterns for body sensor network activity recognition," in *Proc. IEEE Real-Time Syst. Symp. (RTSS)*, 2012, pp. 95–104.
[23] Y. Wu, G. Zhou, and J. A. Stankovic, "ACR: Active collision recovery in dense wireless sensor networks," in *Proc. IEEE 29th Conf. Inf. Commun. (INFOCOM)*, 2010, pp. 911–919.
[24] Y. Li *et al.*, "Communication energy modeling and optimization through joint packet size analysis of BSN and WIFI networks," *IEEE Trans Parallel Distrib. Syst.*, vol. 24, no. 9, pp. 1741–1751, Sep. 2013.
[25] J. Duan, D. Gao, D. Yang, C. Foh, and H.-H. Chen, "An energy-aware trust derivation scheme with game theoretic approach in wireless sensor networks for IoT applications," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 58–69, Feb. 2014.
[26] F. Han *et al.*, "Time-reversal wireless paradigm for green Internet of Things: An overview," *IEEE IoT J.*, vol. 1, no. 1, pp. 81–98, Feb. 2014.
[27] V. K. Ingle and J. G. Proakis, *Digital Signal Processing Using MATLAB*, Stamford, CT, USA: Cengage Learning, 2011.
[28] D. G. Cacuci, *Sensitivity and Uncertainty Analysis: Theory, Volume I*. Boca Raton, FL, USA: CRC Press, 2003.
[29] D. G. Cacuci, M. Ionescu-Bujor, and I. M. Navon, *Sensitivity and Uncertainty Analysis: Applications to Large-Scale Systems, Volume I*. Boca Raton, FL, USA: CRC Press, 2005.
[30] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[31] N. Littlestone and M. K. Warmuth, "The weighted majority algorithm," in *Proc. IEEE FOCS*, 1989, pp. 256–261.



**Zhen Ren** received the Ph.D. degree in computer science from the College of William and Mary, Williamsburg, VA, USA, in 2012.

Her research interest includes wireless communication and networking and sensor networks, especially body sensor networks, ubiquitous computing, quality of service, and voice over Internet protocol.



**Xin Qi** received the B.Sc. degree in computer science from the Nanjing University, Nanjing, China, in 2007, the M.E. degree from the LIAMA, a joint laboratory between the Chinese Academy of Science, Shanghai, China, and INRIA, in 2010, respectively, and is currently working toward the Ph.D. degree in computer science from the College of William and Mary, Williamsburg, VA, USA.

His research interests include ubiquitous computing and mobile systems.

**Gang Zhou** (GSM'06–M'07–SM'13) received the Ph.D. degree from the University of Virginia, Charlottesville, VA, USA, in 2007.

He is currently an Associate Professor with the Computer Science Department, College of William and Mary, Williamsburg, VA, USA. He has authored or coauthored over 50 academic papers in the areas of smartphones and ubiquitous computing, sensor networks, and wireless communication and networking. The total citations of his papers are more than 3800 according to Google Scholar, among which three of them have been transferred into patents and the MobiSys'04 paper has been cited more than 700 times. Ten of his papers have each attracted more than 100 citations.

Dr. Zhou is a Senior Member of the ACM. He is currently on the Editorial Board of the IEEE INTERNET OF THINGS and also Elsevier Computer Networks. He was the recipient of an award for his outstanding service to the IEEE Instrumentation and Measurement Society in 2008. He was the recipient of the Best Paper Award of IEEE ICNP 2010, which was given to only one paper from among the 170 papers submitted (acceptance rate: 18%). He was also the recipient of the NSF CAREER Award in 2013.

**Haining Wang** (S'97–M'03–SM'09) received the Ph.D. degree in computer science and engineering from the University of Michigan, Ann Arbor, MI, USA, in 2003.

He is currently an Associate Professor of computer science with the College of William and Mary, Williamsburg, VA, USA. His research interests includes the areas of security, networking system, and cloud computing.