Resource-Aware Distributed Epilepsy Monitoring Using Self-Awareness From Edge to Cloud

Farnaz Forooghifar, Amir Aminifar, David Atienza
Embedded Systems Laboratory (ESL), Swiss Federal Institute of Technology Lausanne (EPFL), Switzerland
Email:{farnaz.forooghifar, amir.aminifar, david.atienza}@epfl.ch

***Abstract*—The integration of wearable devices in humans' daily lives has grown significantly in recent years and still continues to affect different aspects of high-quality life. Thus, ensuring the reliability of the decisions becomes essential in biomedical applications, while representing a major challenge considering battery-powered wearable technologies. Transferring the complex and energy-consuming computations to fogs or clouds can significantly reduce the energy consumption of wearable devices and result in a longer lifetime of these systems with a single battery charge. In this work, we aim to distribute the complex and energy-consuming machine-learning computations between the edge, fog, and cloud, based on the notion of self-awareness that takes into account the complexity and reliability of the algorithm. We also model and analyze the trade-offs in terms of energy consumption, latency, and performance of different Internet of Things (IoT) solutions. We consider the epileptic seizure detection problem as our real-world case study to demonstrate the importance of our proposed self-aware methodology.**

***Index Terms*—IoT, edge, fog, cloud, distributed health monitoring, self-awareness, epilepsy.**

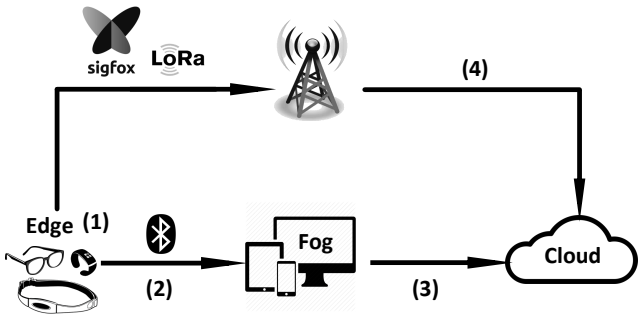

Figure 1: Overview of distributed health monitoring over the Internet of Things (IoT) infrastructure: (1) Edge computing, (2) Edge plus fog computing, (3) Edge plus cloud computing with communication through fog gateway, and (4) Edge plus cloud computing

I. Introduction

Wearable devices are integrated in everyday life of humans, monitoring their activities and analyzing their health conditions using many different sensors [1], [2]. According to the statistics, by 2021,the number of wearable devices will be approximately 929 million, which is a massive increase from the 325 million of 2016 [3]. To guarantee reliable functioning of these devices for real-time health monitoring, a long battery lifetime, and thus, an intelligent energy management technique is required.

The solution recently being considered to manage battery lifetime of wearable devices is the migration of complex and energy-hungry tasks to higher level infrastructures that can provide more computational resources [4]. Different computation layers including fog (personal devices such as cellphones and smart watches) and cloud are available for interaction with wearable devices as shown in Figure 1. Deciding whether to communicate with higher layers depends on the trade-off between communication and computation costs, in order to reduce the overall energy consumption of wearable devices and improve their battery lifetime.

Self-awareness is a promising solution for reducing system's energy consumption and improving battery lifetime of wearable devices. Self-aware systems are equipped with control units which facilitate monitoring their own performance, adapting to changes and improving autonomously [5], [6]. The notion of self-awareness provides the system with knowledge about itself and also its environment and their changes during the time. Thus, the system can adapt to these new situations and predict the effect of dynamic changes to fulfill continuous high performance operation according to the defined goals of the system. In the task distribution over higher computation layers via communication, self-awareness can provide us with information to determine whether this communication can contribute in total energy reduction.

In this article, we analyze workload distribution over edge, fog and cloud in order to minimize the energy consumption of the edge device and enhance its battery lifetime. We study the communication and computation costs of edge device for different workload distribution scenarios, which provides us with the information to obtain the most efficient solution for the particular application we consider. Moreover, we adopt the notion of self-awareness to gain maximum energy saving using task distribution. In order to evaluate our proposed methodology, we consider a real-time epileptic seizure detection system, as our real-world case study.

Epilepsy is one of the most common chronic diseases affecting more than 50 million people worldwide [7]. Despite the recent advances in anti-epileptic drugs, one-third of the epileptic patients still suffer from this disorder. Moreover, epilepsy represents the second neurological cause of years of potential life lost, primarily due to seizure-triggered accidents

Manuscript received July 15, 2019.

and sudden unexpected death in epilepsy (SUDEP) [8]. To be able to notify family members, caregivers, and emergency units in case of a seizure for help, monitoring epileptic patients in real time is necessary. This can help reducing seizure-related injuries, status epilepticus, and SUDEP [9]. Although the gold standard in epilepsy monitoring is based on the video-electroencephalogram, due to its intrusive nature [10], [11], electrocardiogram (ECG) monitoring has recently attracted a lot of attention.

We consider the cost of different workload distribution scenarios for the ECG-based epilepsy monitoring system proposed in [12], [13] and optimize the energy consumption for this target system using self-awareness. The main contributions of this article are as follows:

- The first contribution of this work is distributed epilepsy monitoring over edge, fog, and cloud with the goal of improving the battery lifetime of the edge device. We investigate task distribution between edge and higher level computing infrastructures, including fog and cloud. We model the latency and energy of four main task distribution scenarios, all shown in Figure 1, considering both computation and communication infrastructures. These scenarios include: 1) Edge computing, 2) Edge plus fog computing, 3) Edge plus cloud computing with communication through fog gateway, and 4) Edge plus cloud computing, which are thoroughly analyzed in this article.
- The second contribution of this work is utilizing the notion of self-awareness to perform optimization and select an energy-efficient strategy for distributed health-monitoring between edge, fog and cloud, in order to maximize the battery lifetime of the edge device. Leveraging self-awareness, the heavy computation is distributed over the fog and cloud engines, only when a computationally light-weight low-power learning algorithm on the edge wearable device is not sufficient to make a confident decision about patient's status. Besides, communication with cloud is also done in emergency cases to notify doctors. We validate our technique on an epileptic seizure detection system with the INYU platform [14] using the EPILEPSIA dataset [15], which consists of ECG data from 30 patients with 277 seizures recorded in 4603 hours.

The rest of this article is organized as follows. In Section II, we briefly review the latest studies on task distribution over edge and higher level computation infrastructures and also on the self-awareness in biomedical applications. Section III contains the details on the components of IoT platforms including the computation and communication infrastructures. In Sections IV and V, we formulate the energy and latency of both computation and communication for edge devices in different task distribution scenarios, both including or not the notion of self-awareness. Then, the experimental setup and results are discussed in Sections VI and VII, respectively. In particular, we evaluate the efficiency of our proposed self-aware medical wearable solution in terms of energy, latency and performance against the system without self-aware energy management. Finally, in Section VIII, we summarize the main conclusions of this article.

II. State of the art

In this section, we review the recent studies in distributed biomedical health monitoring systems over the edge, fog and cloud. In addition, we review the recent studies, which leverage the notion of self-awareness for performance enhancement and energy optimization.

A. Edge, Fog, and Cloud in Biomedical Domain

Several studies have recently addressed task distribution over wearable devices and higher computation layers. A survey on cloud-based processing for health-monitoring is done in [16]. One of the main benefits of moving computations on cloud is long-term storing of patients' bio-signals for better analysis of their health condition and its changes during long periods. The concept of cloudlet computing is also analyzed as a platform to run time-critical tasks. Storing data in clouds also removes the necessity to repeat the same test in different hospitals for the same patient [17]. In [18], the challenge of security of the communication with higher layers is addressed using water mark and user identification codes. Another issue in using higher computation levels is management of huge amount of medical data which is addressed in [19], where the scheduling of virtual machines in cloud environments is improved using parallel processing.

In [20], [21], the fog layer is introduced as a third level of computation between edge and cloud for healthcare IoTs, which could achieve more than 90% bandwidth efficiency in their ECG feature extraction case study. In [22], Convolutional neural network is implemented in fog layer which is again considered in ECG classification and impressive reduction in transmission time is achieved replacing cloud with the fog layer. In [23], the authors propose a concept architecture for real-time remote cardiac health-monitoring with long QT syndrome case study. They use ZigBee to communicate with cloudlet for resource-intensive tasks with context-aware concentration of data. In [24], they have taken into account the battery status of the wearable sensors to put the system in a low-power monitoring mode in case of battery shortage. The fog level is used in this paper to do more complex tasks such as state detection. However, none of these works have considered cloud as the third layer to handle higher complexity tasks or to share the information with the doctors.

The authors of [25] have developed an Android app to monitor the ECG signal of patients and save it in private cloud server to be retrieved by medical personnel for analysis. As a result, their main focus is on sending the data efficiently using compression and also guaranteeing the security of the platform by applying encryption methods. Thus, they do not do energy and latency modeling and do not perform any distributed pathology detection over cloud. In [26], the connection between wearable sensors and cloud infrastructure is provided by personal servers, which is used for basic analysis and aggregation. They have considered congestive heart failure as the case study to analyze their proposed system. However, the

authors do not consider dynamic distribution of workload over the edge, fog, and cloud infrastructure using the self-awareness concept. Then, it is shown in [27], that for their platforms, which are smartglasses and smartwatches, the overall energy consumption and communication delay are reduced with direct internet connection via WiFi compared to using Bluetooth. Nevertheless, the authors do not formulate the end-to-end latency and energy consumption and do not consider the self-aware distribution of workload over the edge, fog, and cloud infrastructure.

The analysis of Mobile Cloud Computing (MCC) is done in [28], using analytical modeling where the energy and delay trade-offs are discussed. They have discussed different scenarios of task offloading to smartphone and cloud analyzing delay and power consumption of using WiFi and LTE standards. Although they have provided energy and latency formulation for both computation and communication between wearable system, smart phone and cloud, their modeling do not contain any energy-management technique. In [29], the best computation migration scenario is selected to optimize the latency of the system in arrhythmia classification. However, the authors have not considered the energy consumption and lifetime of such edge devices. Moreover, they do not leverage the notion of self-awareness to distribute workload over the fog/cloud infrastructure with higher processing power. In [30], a high-quality and low-power cardiovascular monitoring system is proposed which communicates with cloud using Bluetooth Low Energy (BLE) and LoRA. This work provides a comparison between BLE and LoRa, but does not provide a general formulation for the optimization of energy and latency in different distribution scenarios. In conclusion, the previous studies do not consider a general formulation of dynamic distribution of workload over the edge, fog, and cloud infrastructure using the self-awareness concept, taking into account both the real-time operation or end-to-end latency and the energy consumption or battery lifetime.

B. Self-Awareness in Biomedical Applications

Self-awareness has been one of the promising concepts to improve system's performance and reduce its energy consumption in literature [31]–[37]. In health monitoring this concept is used to reduce energy consumption and extend the battery lifetime of wearable devices. In [38], remote health monitoring is performed based on personalized data (such as age, gender, etc.) and situation-awareness is adopted to increase the accuracy of remote health monitoring. In addition, different priorities are given to the sensory data collection to consider the energy efficiency and dependability of the system.

In [39], different observation parameters such as confidence and history are described and a high-quality description of the system from raw data using these parameters is provided for an emotion recognition system. The authors of [40] proposed data confidence evaluation system that analyzes the data gathered from sensors to evaluate their reliability. In [13], a self-aware seizure detection system is proposed where both energy reduction and performance improvement are obtained using confidence evaluation. In [41], automatic labeling of seizure in epilepsy detection system is performed using self-learning.

In conclusion, there is no global optimum scenario for all health monitoring applications. Thus, in order to enable distributed health monitoring over edge, fog and cloud, the latency, energy, and lifetime of the system should be analyzed for each particular case study. Therefore, in this article we propose a model to estimate the energy and latency of both computations and communications that should be done by the edge device plus the energy consumed by the fog layer in different task distribution scenarios over edge, fog and cloud. Our system also benefits from the notion of self-awareness to adopt the most efficient scenario among all to minimize energy consumption. This is fulfilled by managing multiple levels of computation over edge, fog and cloud. Different scenarios are then applied on an ECG-based seizure detection system to build a distributed epilepsy monitoring system using self-awareness over edge, fog, and cloud.

III. Internet of Things (IoT) Infrastructure

In this section, we provide the background for computation and communication infrastructures in IoT platforms. The high-level overview of an IoT platform is shown in Figure 1, including the edge sensors, the fog devices and the cloud engines. Four different distribution scenarios are considered for this framework: 1) performing all the tasks on the edge sensors, 2) task distribution between the edge sensors and the fog devices, 3) task distribution between the edge sensors and the cloud engines using the fog devices as gateways, and 4) direct task distribution between the edge sensors and the cloud engines.

A. Computation Infrastructure

- **Edge:** These local devices are equipped with bio-sensors and perform multi-signal acquisition, which results in significant energy consumption even without considering other processes. Although using edge sensors for light computational tasks results in the lowest response time, in order to enable them to operate for days with single battery charge, the energy-hungry computations are often distributed to higher computation layers, e.g., fog and cloud engines, which are equipped with more computational resources.
- **Fog:** These devices are usually within short distances from the edge devices and, as a result, the latency overhead of communication with them is not very high. On the other hand, offloading the energy-hungry computations on the fog devices reduces the computation overhead on the edge devices. Despite the aforementioned advantages, still, such devices are limited in terms of computational power and energy consumption to run very complex machine learning and signal processing techniques.
- **Cloud:** The cloud engines are the most computationally powerful platforms and provide the illusion of infinite computational resources, with no energy limitation. However, as cloud engines are at long distances from the users and edge devices, the latency and energy of communication are substantial and become the main

Table I: Notation used for modeling response-time latency and energy consumption

Notation	Description	Notation	Description
L_{Edge}	edge computation latency	E_{Edge}	edge computation energy
L_{AC}	acquisition latency	E_{AC}	acquisition energy
L_{PP}	pre-processing latency	E_{PP}	pre-processing energy
L_{FE}	feature extraction latency	E_{FE}	feature extraction energy
L_{ML}	machine learning latency	E_{ML}	machine learning energy
$L_{E\rightarrow F,TX}$	edge-fog transmission latency	$E_{E\rightarrow F,TX}$	edge-fog transmission energy
L_{TX}	latency of transmission using BLE	E_{idle}	edge idle energy
L_{RX}	latency of receive using BLE	$E_{E\rightarrow F,RX}$	edge-fog receive energy
$L_{E\rightarrow C,TX}$	edge-cloud transmission latency	$E_{E\rightarrow C,TX}$	edge-cloud transmission energy
$L_{FE}(i)$	i^{th} level feature extraction latency	$E_{FE}(i)$	i^{th} level feature extraction energy
$L_{ML}(i)$	i^{th} level machine learning latency	$E_{ML}(i)$	i^{th} level machine learning energy
$L_{CF(i)}$	i^{th} level confidence calculation latency	$E_{CF(i)}$	i^{th} level confidence calculation energy
γ_F	fog computation speed-up factor	$V_{E\rightarrow F}$	data volume transmitted from edge to fog
B_{BLE}	BLE bandwidth	BER_F	bit error rate in communication with the fog
I_{TX}	current of transmission using BLE	V_{TX}	voltage of transmission using BLE
I_{RX}	current of receive using BLE	V_{RX}	voltage of receive using BLE
γ_C	cloud computation speed-up factor	$V^{i,j}$	volume of data transmitted between i^{th} and j^{th} data centers
$B_E^{i,j}$	communication bandwidth between i and j	B_l^j	local bandwidth of destination j
$D^{i,j}$	distance between i^{th} and j^{th} data centers	S_l	speed of light
BER_C	bit error rate in communication with the cloud	E_{packet}	energy of transmitting one packet
N_{packet}	number of transmitted packets	S_{packet}	size of each packet
Len_{TX}	length of transmitted data	Len_H	length of header
l	number of levels in self-aware system	P_i	probability of invoking i_{th} classifier
$t-1, t_1-1$	number of levels calculated on the edge	t_2-t_1	number of levels calculated on the fog in $E \xrightarrow{\text{F}} C$

limitation. Although the communication energy forced on edge devices can be reduced by performing a two-phase communication, i.e., first, between the edge and fog and, then, between the fog and cloud, the long latency still remains a significant drawback.

B. Communication Infrastructure

- **Bluetooth Low Energy (BLE):** To communicate between edge and fog, we use BLE protocol [42], which is a low-power protocol suitable for short distance communications. We consider common rate of transmission for BLE, which is 1 Mbps. To calculate the power, we also adopt the values provided by Nordic DevZone online power analyzer [43]. It assumes that the transmission current is consumed by the system during transmission and for the rest of the times the idle current is consumed. Then, the average current and the power are calculated according to this assumption.
- **Sigfox:** This is a light-weight protocol used for sending small amount of data with low power consumption. We can send 6 messages per hour (12 bytes) and thus, 144 messages per day containing 140 uplink and 4 downlink [44]. This protocol is energy-efficient as it consumes only 15 to 45 mA during a few seconds (6s per message) and the idle consumption is negligible. As discussed, according to its low communication capacity per day only small data or messages can be transmitted through this protocol.
- **LoRa:** This protocol follows fair access policy that limits the uplink air time to 30 seconds per day (24 hours) per node. For 10 bytes of payload, this translates in 20 messages per day at SF12 or 500 messages per day at SF7. The downlink messages are limited to 10 messages per day (24 hours) per node. A reliable goal is to keep the application payload under 12 bytes, and the interval between the messages at least several minutes [45].
- **WiFi:** The communication with cloud is done using WiFi standard considering several data centers in between. Up to 100 Mbps can be obtained from this protocol depending on the traffic [46]. Also, continuous transmission of data is possible provided that there is an access to the network.

IV. Modeling of Energy and Latency for the Edge Sensor in Distributed Health Monitoring

In this section, we formulate the energy consumption and response-time latency of different distribution scenarios, which will then be used to optimize the system and improve battery lifetime in wearable health monitoring devices. All notations used in this section and the next section can be found in Table I.

A. Edge Formulation

In this part we describe the scenario where there is no communication between the edge and the higher layers.

- **Latency**: This is the latency of performing different training phases of the system which includes the steps shown in Figure 2 in our epilepsy detection case study. The latency of the communication (L_{Edge}) is calculated as:

$$L_{Edge} = L_{AC} + L_{PP} + L_{FE} + L_{ML}, \qquad (1)$$

where L_{AC}, L_{PP}, L_{FE}, and L_{ML} are the latencies for acquisition of sufficient number of samples to be processed by the signal processing and machine learning

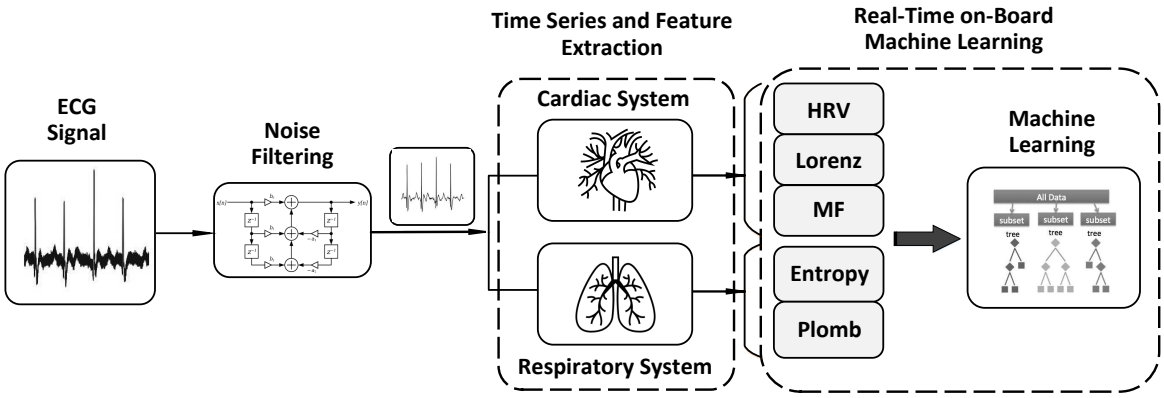

Figure 2: Overview of the epileptic seizure detection system used as our case study [13]

algorithms, preprocessing, feature extraction and machine learning, respectively.

- **Energy**: This energy consists of the energies for performing different steps in the learning procedure (Figure 2 as an example):

$$E_{Edge} = E_{AC} + E_{PP} + E_{FE} + E_{ML} + E_{idle}, \quad (2)$$

where E_{AC}, E_{PP}, E_{FE}, and E_{ML} are the energies due to acquisition of sufficient number of samples to be processed by the signal processing and machine learning algorithms, preprocessing, feature extraction and machine learning, respectively. We also consider the E_{idle}, which represents the energies consumed in the idle phase, containing leakage energy as well as the energy from different power saving states of the edge device.
In the latest low-power sensor nodes, which do not need to update memory regularly, thanks to advanced power management techniques [47], the system is in the sleep mode during the idle phase and only a small subsystem monitors the state of the system to wake up the entire system when there is pending processing needed to be done. Therefore, we assume that for such systems, the idle energy is almost equal to the sleep energy, with negligible energy spent in sleep states, compared to the computation energy.

B. Edge→Fog Formulation

In this section, we formulate the scenario when the edge sensor communicates with the fog device and complex calculations are handled by the fog layer instead of the edge sensor.

- **Latency**: This is the latency calculated in Eq. (1) plus the latency of communication between the edge and the fog.

$$L_{Edge}(E \to F) = L_{AC} + L_{PP} + \gamma_F \cdot (L_{FE} + L_{ML}) + L_{E\to F,TX}, \quad (3)$$

where $L_{E\to F,TX}$ is the transmission latency. We assume that the most complex tasks, which are the feature extraction and machine learning, are performed by the higher layer, where γ_F is the speed-up factor at the fog. As the fog devices include high-performance computing resources, the computation time is reduced and we have $\gamma_F \leq 1$.
To communicate between the edge and the fog, we use the BLE and, in order to compute the latency, we calculate the amount of data we transfer. As a result, the formula is:

$$L_{E\to F,TX} = \frac{V_{(E\to F)} \cdot (1 + BER_F)}{B_{BLE}}, \quad (4)$$

where $V_{(E\to F)}$ is the volume of data being transferred between the edge and the fog and B_{BLE} is the bandwidth of the BLE, which is commonly 1 Mbps. For our epilepsy detection system, the sampling frequency is 256 Hz and we assume that each data sample is 4 Bytes. As a result, the entire amount of data being transferred in one second is 8192 bits, which is equal to 8.192 ms of latency. The term BER_F is bit error rate, which is the probability of missing a bit being transmitted from the edge device to the fog layer.

- **Energy**: In this case, the total energy is:

$$E_{Edge}(E \to F) = E_{AC} + E_{PP} + E_{E\to F,TX}. \quad (5)$$

Compared to Eq. (2), the E_{FE} and E_{ML} are removed from the computation energy of the edge, as they are performed on the fog instead of the edge device. Besides the energy of the edge sensors, the energy consumption of the fog device should also be modeled, as it also has a limited source of energy:

$$E_{Fog}(E \to F) = E_{FE} + E_{ML} + E_{E\to F,RX}. \quad (6)$$

To calculate the communication energy between the edge and the fog, which is done using BLE, we have extracted the values from Nordic DevZone power estimator [43], with the following formulas:

$$E_{E\to F,TX} = I_{TX} \cdot L_{TX} \cdot V_{TX} = I_{TX} \cdot L_{E\to F,TX} \cdot V_{TX}, \quad (7)$$

$$E_{E\to F,RX} = I_{RX}\cdot L_{RX}\cdot V_{RX} = I_{RX}\cdot L_{E\to F,TX}\cdot V_{RX}, \quad (8)$$

where I_{TX} and L_{TX} are the average current and latency of transmission and V_{TX} is the voltage of transmission between the edge and the fog. Similarly, I_{RX}, L_{RX}, and V_{RX} are the average current, latency, and voltage of the receiver in the fog device. The latency of the transmission is obtained from Eq. (4).

C. Edge→Cloud Formulation

In this section, we model the scenario in which the edge device communicates with the cloud and complex calculations are performed by the cloud engine instead of the edge device.

- **Latency**: The total latency is composed of the computation latency (Eq. (1)) and the communication latency between edge and cloud, namely:

$$L_{Edge}(E\to C) = L_{AC} + L_{PP} + \gamma_C\cdot(L_{FE}+L_{ML}) + L_{E\to C,TX}. \quad (9)$$

Similar to the fog layer the computation time is reduced on the cloud engine and we have $\gamma_C \leq 1$. We adopt the latency model proposed in [48] and consider a fully-connected network with full-duplex peer-to-peer global optical fiber links among the data centers. Moreover, for a fast data transfer, we use an all-bandwidth policy, with predetermined reserved bandwidth for communication among each two data centers. Therefore, it is sufficient to consider the source i and the destination data center j, in our analysis, connected to the edge sensor and cloud, respectively. The communication latency between the edge sensor and the cloud engine is then, calculated as follows [48]:

$$L_{E\to C,TX} = \frac{V^{i,j}}{B_l^j} + \frac{V^{i,j}\cdot(1+BER_C)}{B_E^{i,j}} + \frac{D^{i,j}}{S_l}. \quad (10)$$

The first term correspond to the local latency of the destination data center, where $V^{i,j}$ is the volume of data being transferred from i to j and B_l^j is the local bandwidth of j. The second term captures the transmission latency of the network, where $B_E^{i,j}$ is the bandwidth reserved for communication between source i and destination j. The bit-error-rate is captured by BER_C in the transmission latency and is used to obtain the effective communication bandwidth between the source and destination data centers. Finally, the last term captures the propagation latency, where $D^{i,j}$ is distance from i to j and S_l is the speed of light.

- **Energy**: In this case, the total energy is the energy consumed for calculation on the edge device plus the energy of communication with cloud ($E_{E\to C,TX}$):

$$E_{Edge}(E\to C) = E_{AC} + E_{PP} + E_{E\to C,TX}. \quad (11)$$

We consider LoRa [45] to communicate between the edge device and the cloud engine. The transmission energy for this protocol can be calculated as:

$$E_{E\to C,TX} = E_{packet}\cdot N_{packet} = E_{packet}\cdot\frac{Len_{TX}+Len_H}{S_{packet}}, \quad (12)$$

where E_{packet} and N_{packet} are the energy of transferring one packet and the number of packets, respectively. The number of packets is calculated as the length of the transmitted data (Len_{TX}) plus the header length (Len_H) of minimum 13 bytes divided by the size of each packet (S_{packet}).

D. Edge $\xrightarrow{Fog}$ Cloud Formulation

In this section, the formulas of the latency and energy in the case of communication between the edge device and the cloud engine through the fog layer are calculated.

- **Latency**: In case of latency, this communication is almost equal to the sum of communication of the edge with the fog and with the cloud (as the distance of the edge to the fog is negligible compared to the distance with the cloud). As a result, for the latency we have:

$$L_{Edge}(E\xrightarrow{F} C) = L_{AC} + L_{PP} + \gamma_C\cdot(L_{FE}+L_{ML}) + L_{E\to F,TX} + L_{E\to C,TX}, \quad (13)$$

where $L_{E\to F,TX}$ and $L_{E\to C,TX}$ are calculated as in Eq. (4) and Eq. (10).

- **Energy**: In this case, we first transfer the data from the edge to the fog and then from the fog to the cloud. As a result, the wearable device is only involved with the first part of transmission to the fog and the energy consumed for communication will be $E_{E\to F,TX}$ and thus, the total energy is calculated as follows:

$$E_{Edge}(E\xrightarrow{F} C) = E_{AC} + E_{PP} + E_{E\to F,TX}, \quad (14)$$

which is the same as Eq. (5). In this case, for the fog layer we obtain:

$$E_{Fog}(E\xrightarrow{F} C) = E_{E\to F,RX} + E_{F\to C}. \quad (15)$$

V. Self-Aware Distribution of Machine Learning Algorithm

In this section, we first describe briefly our self-aware energy management technique. Then, we extend the response-time latency and energy consumption models developed in Section IV by introducing the notion of self-awareness in such systems.

A. Self-Aware Classification Algorithm

To reduce the energy consumption without sacrificing the quality of the classification, in our previous work [13], we proposed a two-level self-aware classifier. There, we have developed a self-aware seizure detection technique where classification can be done with either a simple set of features

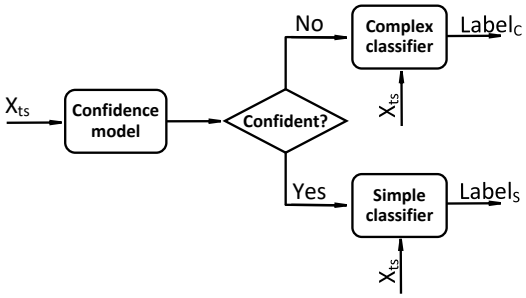

Figure 3: Test phase of the self-aware energy management proposed in [13]

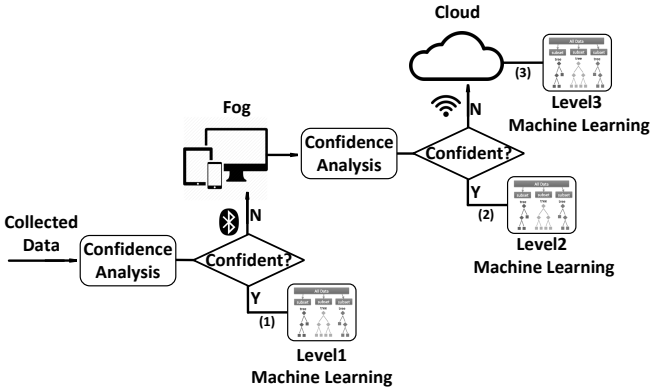

Figure 4: Task distribution scenario over edge, fog and cloud using the notion of self-awareness and concept of confidence

or a more complex set. In fact, the entire set of features are not used for seizure detection unless confident classification based on the set of simple features is not possible. Then, we take advantage of the multi-mode execution possibilities of the platform, in a self-aware fashion, thus the energy consumption is reduced while the quality of system remains in an acceptable level for medical use. Moreover, our system is kept in an ultra-low power (energy-saving) mode when tasks terminate their execution.

Moreover, we extend this classification algorithm to multiple levels with several classifiers, with different detection performances and energy consumptions. The system starts with the simplest classifier, namely, with the minimum energy consumption and detection performance. If the result is not deemed reliable, it invokes the next simplest classifier and continues this procedure until the decision is reliable. In this technique, to reach the maximum energy saving, the system has to be aware of the detection quality that it can provide in each level. To achieve this, we adopt the self-awareness concept and introduce the notion of confidence to investigate whether the decision of a classifier is reliable.

Several previous studies consider efficient implementation of neural networks over resource-constrained edge IoT platforms and embedded systems to improve energy efficiency [49]–[55]. On the other hand, our proposed technique in this article is complementary to the solutions proposed for the implementation of neural networks in [49]–[55], i.e., the previous techniques can be adopted together with our proposed technique in this article, demonstrating the generality of our proposed self-aware solution.

The test phase in case of the two-level classifier is shown in Figure 3, where the confidence model, which is extracted in the train phase, decides whether the simple model is confident to classify the new data (X_{ts}). If so, the classification is done by the simple classifier (with the final decision $Label_S$); otherwise, the complex classification is invoked (with the final decision $Label_C$). Combining this technique with our distributed health monitoring approach, as shown in Figure 4, only in the case that the complex classification is required to ensure the reliability of the decision, we offload the complex computation on the fog and cloud engines. We also consider a threshold for the confidence, which shows the minimum number of trees (out of 100) in the random forest algorithm that should agree on using the simple classifier. By increasing this threshold, the frequency of decisions that are made by the simple classifier is reduced, which leads to an increase in the overall classification performance. On the other hand, higher thresholds increase the overall complexity and the energy consumption of the proposed classification algorithm.

B. Edge Formulation

In this section, we consider the case when no communication is done between the edge device and the fog and cloud engines.

- **Latency**: As in our self-aware system, we consider several levels of classification. First, we calculate the latency of each of these levels, as follows:

$$L_i = L_{FE(i)} + L_{ML(i)}, \quad i = 1, \ldots, l, \tag{16}$$

where L_i is the total latency of i^{th} level and $L_{FE(i)}$ and $L_{ML(i)}$ are the latencies of i^{th} level feature extraction and machine learning, respectively. The total number of levels is equal to l.
Based on the probability of invoking the i^{th} classifier, denoted by P_i, the total latency of computation on the edge device for our wearable system is calculated as follows:

$$\begin{aligned} L_{SA}(Edge) = L_{AC} + L_{PP} + \sum_{i=1}^{l} P_i \cdot L_i \\ + L_{CF(1)} + \sum_{i=2}^{l} (1 - \sum_{j=1}^{i-1} P_j) \cdot L_{CF(i)}, \end{aligned} \tag{17}$$

where $L_{CF(i)}$ is the latency of calculating i^{th} level confidence. Then, second line of this equation indicates the total latency of confidence calculation. As demonstrated, we always calculate the confidence of first layer and for the next layers we just calculate the confidence if a classifier of lower level is not chosen yet (the probability is calculated as $1 - \sum_{j=1}^{i-1} P_j$).
- **Energy**: Similar to the latency, we calculate the energy of each level as follows:

$$E_i = E_{FE(i)} + E_{ML(i)}, \quad i = 1, \ldots, l, \tag{18}$$

where $E_{FE(i)}$ and $E_{ML(i)}$ are the energies of the i^{th} level feature extraction and machine learning, respectively. Then, for the entire computation on the edge, we have:

$$\begin{aligned} E_{SA}(Edge) = E_{AC} + E_{PP} + \sum_{i=1}^{l} P_i \cdot E_i \\ + E_{CF(1)} + \sum_{i=2}^{l}(1 - \sum_{j=1}^{i-1} P_j) \cdot E_{CF(i)}, \end{aligned} \tag{19}$$

where $E_{CF(i)}$ is the energy of calculating i^{th} level confidence.

C. Edge→Fog Formulation

In this section, we consider the scenario when the edge sensor communicates with the fog device and the complex calculations are performed on the fog, instead of the edge sensor. This is done when the system needs the classifiers with levels higher than $t-1$, which is selected according to the available resources, on the edge sensor.

- **Latency**: The communication is done only when we need classification of levels equal or higher than t, which are the energy-hungry tasks. According to Eq., (3) we have the following latency for our self-aware system:

$$\begin{aligned} L_{SA}(E \to F) = L_{AC} + L_{PP} + \sum_{i=1}^{t-1} P_i \cdot L_i \\ + \sum_{i=t}^{l} P_i \cdot (L_{E\to F,TX} + \gamma_F \cdot L_i) \\ + L_{CF(1)} + \sum_{i=2}^{t-1}(1 - \sum_{j=1}^{i-1} P_j) \cdot L_{CF(i)} \\ + \sum_{i=t}^{l}(1 - \sum_{j=1}^{i-1} P_j) \cdot \gamma_F \cdot L_{CF(i)}, \end{aligned} \tag{20}$$

where $L_{E\to F,TX}$ is calculated as in Eq. (4).
The first two terms in Eq. (20) are latencies of acquisition and pre-processing steps for the entire signal. Then, the latency of the levels of classifications that are performed on the edge device is calculated, which is equal to multiplying probability of each level with its latency. For level t and higher we have the latency of transmission from the edge device to the fog layer ($L_{E\to F,TX}$) and also the latency of computation multiplied by fog speed-up factor (γ_F). The third line of the equation corresponds to the confidence levels that are calculated on the edge device, and the last line is the latency of calculating these confidence levels in the fog device.

- **Energy**: Similar to the latency, according to Eq. (5), we obtain:

$$\begin{aligned} E_{SA}(E \to F) = E_{AC} + E_{PP} + \sum_{i=1}^{t-1} P_i \cdot E_i \\ + \sum_{i=t}^{l} P_i \cdot E_{E\to F,TX} \\ + E_{CF(1)} + \sum_{i=2}^{t-1}(1 - \sum_{j=1}^{i-1} P_j) \cdot E_{CF(i)}, \end{aligned} \tag{21}$$

where $E_{E\to F,TX}$ is calculated as in Eq. (7).
Here, the first two terms are acquisition and preprocessing energy of the entire signal and then we have energy of classifications, which are done on the edge device. In the second line we calculate transmission energy for the level t and higher and in the last line the energy of calculating confidence of levels lower than t is considered.

D. Edge→Cloud Formulation

In this section, the edge communicates with the cloud in case of classification of layers t and higher and energy-hungry calculations are performed by the cloud engine instead of the edge sensors.

- **Latency**: According to Eq. (9), we have the following latency:

$$\begin{aligned} L_{SA}(E \to C) = L_{AC} + L_{PP} + \sum_{i=1}^{t-1} P_i \cdot L_i \\ + \sum_{i=t}^{l} P_i \cdot (L_{E\to C,TX} + \gamma_C \cdot L_i) \\ + L_{CF(1)} + \sum_{i=2}^{t-1}(1 - \sum_{j=1}^{i-1} P_j) \cdot L_{CF(i)}, \\ + \sum_{i=t}^{l}(1 - \sum_{j=1}^{i-1} P_j) \cdot \gamma_C \cdot L_{CF(i)}, \end{aligned} \tag{22}$$

where $L_{E\to C,TX}$ is calculated as in Eq. (10), taking into account the limitation of the communication protocol to obtain the effective bandwidth. The terms are defined in the same order as Eq. (20).
- **Energy**: According to Eq. (11), we have the following energy:

$$\begin{aligned} E_{SA}(E \to C) = E_{AC} + E_{PP} + \sum_{i=1}^{t-1} P_i \cdot E_i \\ + \sum_{i=t}^{l} P_i \cdot E_{E\to C,TX} \\ + E_{CF(1)} + \sum_{i=2}^{t-1}(1 - \sum_{j=1}^{i-1} P_j) \cdot E_{CF(i)}, \end{aligned} \tag{23}$$

where $E_{E\to C,TX}$ is calculated as in Eq. (12). The terms are defined with the same order as Eq. (21).

E. Edge $\xrightarrow{Fog}$ Cloud Formulation

In this section, the formulas of latency and energy in case of communication between the edge sensor and the cloud engine through fog layer are calculated.

- **Latency**: According to Eq. (13), we have the following latency:

$$\begin{aligned} L_{SA}(E \xrightarrow{\text{F}} C) = {} & L_{AC} + L_{PP} + \sum_{i=1}^{t_1-1} P_i \cdot L_i \\ & + \sum_{i=t_1}^{t_2-1} P_i \cdot (L_{E\to F,TX} + \gamma_F \cdot L_i) \\ & + \sum_{i=t_2}^{l} P_i \cdot (L_{E\to F,TX} + L_{E\to C,TX} + \gamma_C \cdot L_i) \\ & + L_{CF(1)} + \sum_{i=2}^{t_1-1} (1 - \sum_{j=1}^{i-1} P_j) \cdot L_{CF(i)} \\ & + \sum_{i=t_1}^{t_2-1} (1 - \sum_{j=1}^{i-1} P_j) \cdot \gamma_F \cdot L_{CF(i)} \\ & + \sum_{i=t_2}^{l} (1 - \sum_{j=1}^{i-1} P_j) \cdot \gamma_C \cdot L_{CF(i)}. \end{aligned} \tag{24}$$

  Here, we assume that first $t_1 - 1$ levels of classification are done on the edge device, then from classification t_1 to $t_2 - 1$ are performed on the fog layer, and the rest of the levels are calculated on the cloud engine. This formulation results in adding two more terms compared to Eq. (22), which correspond to the latency of classifiers on the fog layer (second line) as well as the latency of confidence calculation on the fog layer (5^{th} line).
- **Energy**: According to Eq. (14), we have the following energy:

$$\begin{aligned} E_{SA}(E \xrightarrow{\text{F}} C) = {} & E_{AC} + E_{PP} + \sum_{i=1}^{t_1-1} P_i \cdot E_i \\ & + \sum_{i=t_1}^{l} P_i \cdot E_{E\to F,TX} \\ & + E_{CF(1)} + \sum_{i=2}^{t_1-1} (1 - \sum_{j=1}^{i-1} P_j) \cdot L_{CF(i)}. \end{aligned} \tag{25}$$

  The terms are defined with the same order as Eq. (21).

VI. Experimental Setup

In this section, we discuss the experimental setup for the evaluation of our proposed technique in terms of latency, energy consumption, and classification performance. We consider the epileptic seizure detection problem, a real-world problem, as our case study.

A. Datasets

The proposed distributed epilepsy monitoring approach is evaluated with the EPILEPSIAE dataset [15], which is the largest epilepsy dataset manually annotated by doctors for seizure detection and prediction worldwide and enables us to rigorously evaluate our proposed methodology. This dataset consists of one-lead ECG and 19-channel EEG data. The recordings are made in a routine clinical environment, so nonseizure activity and artifacts such as head/body movement, chewing, blinking, early stages of sleep, and electrode pops/movement are present. No constraints regarding the types of seizure are imposed; the dataset contains complex partial (CP), simple partial (SP), and secondarily generalized seizures (GS) [56].

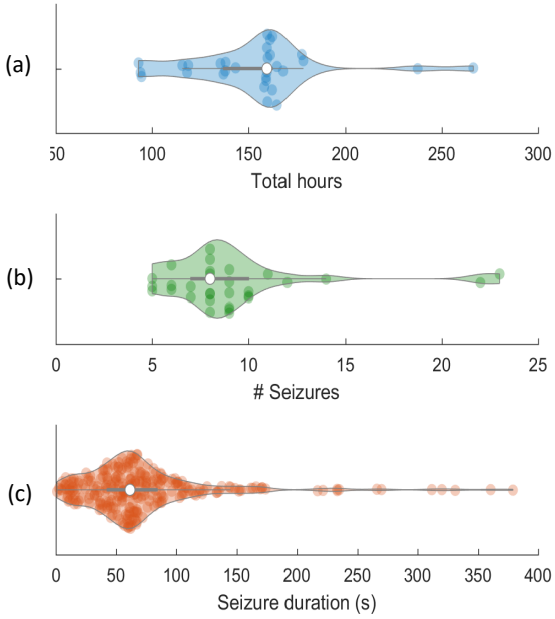

Figure 5: Dataset information including (a) total number of hours for different patients, (b) number of seizures for different patients, and (c) duration of seizures for different patients in seconds

We have used the ECG data of 30 patients with 4603 hours of recordings separated to one-hour files containing 277 seizures. The data is acquired at a sampling rate of 256 Hz with 16-bit resolution. The number of seizures among different patients differs from 5 to 23 seizures per patient with an average of 9.23 seizures per patient. The average duration of these seizures is 75.81 seconds. The total recording duration per patient differs from 92.90 to 266.36 hours, with an average of 153.43 hours. The details of this dataset is also presented in Figure 5.

B. Performance Metrics

As the Figure 5 shows, the seizures are not distributed evenly among the patients. As a result, in order to have a reliable analysis of the detection system for different modes of classification, in each iteration of testing the system 70% of the data is randomly picked as the training data and the rest is picked as the test data. The performance of the proposed algorithm is evaluated by measuring the specificity ($Spec$),

(a) INYU platform (b) INYU sensor and placement

Figure 6: Experimental setup including: (a) the INYU hardware board and platform and (b) the INYU touch and thoracic sensors

sensitivity (Sen), and the geometric mean ($Gmean$), which are defined as follows:

$$Spec = \frac{TN}{FP + TN}, \tag{26}$$

$$Sen = \frac{TP}{TP + FN}, \tag{27}$$

$$Gmean = \sqrt{Spec \cdot Sen}, \tag{28}$$

where FP, TN, TP and FN definitions are the following ones:

- False positive (FP): The patient is in the inter-ictal state, but the sample is classified as ictal.
- True negative (TN): The patient is in the inter-ictal state, and the algorithm declared this situation.
- True positive (TP): The patient is in the ictal state, and the algorithm detected this state.
- False negative (FN): The patient is in the ictal state, and the sample is not classified correctly.

The geometric mean $Gmean$ is adopted since its high values reflect that both specificity ($Spec$) and sensitivity (Sen) are high, which is equal to high quality detection. Conversely, if the geometric mean $Gmean$ is low, then $Spec$, Sen, or both are low, which is undesirable. Finally, we include the geometric mean, as it is the only correct average of normalized measurements, according to [57].

C. Implementation Platform

The target hardware platform for our system is the SmartCardia INYU wearable sensor [14], which is consistent with the standard signal acquisition equipment in hospitals, to evaluate the complexity of proposed solution and battery lifetime of the device. This is done by porting the entire machine-learning algorithms on the INYU device, which includes an ARM Cortex-M3 chipset (STM32L151RDT6) [58] as its processor for data analysis and classification, which is a low-power 32-bit microcontroller with 48 kB RAM and 384 kB flash storage and the maximum frequency of 32 MHz. This processor has several power-management modes, including active and sleep modes, with the possibility of dynamically switching between different modes. The INYU device is powered by a 710 mAh battery, which is used as reference to calculate the relative energies in Table II. The ECG signal acquisition is done using silver-chloride electrodes for impedance pneumography [59]. The analog-to-digital converter (ADC) is the ADS7142 module [60], which is an event-driven ADC. This ADC has a low power consumption of 900 nW and works with 0.5 uA current.

VII. Experimental Results

In this section, we evaluate the efficiency of our proposed distributed epilepsy monitoring system, in terms of latency, energy consumption, and classification performance. For simplicity's sake we assume that $\gamma_F = \gamma_C = 1$ and $BER = 0$. We consider the maximum distance of two points on the earth for the $D^{i,j}$ in cloud communication formulation. Moreover, our detection system consists of two classification levels ($l = 2$) and the simpler level is always performed on the edge device ($t = 2$).

Table II shows the energy consumption of two different scenarios considering both self-aware system with different confidence thresholds and the system without self-awareness. These two scenarios are first, computing completely on the edge device and second, communicating with the fog level. The energies are normalized with respect to the INYU battery power (710 mAh) in this table and also in Table III. The results show that the energy reduction of 13-21% (13%, 21%, 19%, 17%, 15%) is achieved by adopting distributed epilepsy monitoring techniques that exploit the heterogeneous computing and communication infrastructure. Also, in both scenarios, by using self-awareness, the energy is reduced by 81-49% compared to using the complex classifier all the time. This table also contains detection performance of the system with and without self-awareness, which shows that the reduction in the geometric mean due to adopting self-awareness in the system is only 1.22% to 4.58%, when the confidence threshold is reduced from 90% to 60%. As a result, combining the notion of self-awareness with distributed epilepsy monitoring results in significant improvements in battery lifetime with negligible reduction in performance.

Table III summarizes the performance, latency, and energy consumption of different design solutions. Among these solutions, the most energy-efficient choice is to offload the computationally-complex tasks to the fog. This solution reduces the energy consumption substantially with only a negligible communication latency overhead. The energy overhead of communication with the cloud engine through the fog is the same as communication of the wearable device with the fog. However, as the communication with the cloud is done using WiFi, the latency increases about 1.213 seconds for each 60-second window. Therefore, this solution is not efficient in terms of end-to-end latency.

We summarize this discussion for our epileptic seizure detection system in the following:

- The communication with the fog requires the lowest energy (3.65 mJ) and the latency overhead is only approximately 10.4% of the entire end-to-end latency. Therefore, we transfer the data to the fog layer via BLE and receive back the outcome of the complex classification performed on the fog device, which improves the overall performance.

Table II: Summarizing the energy consumption of both computation and communication in different scenarios plus the detection performance of the system with and without self-awareness. The energies are normalized with respect to the INYU battery power (710 mAh).

Scenario	Confidence %	Spec %	Sens %	Gmean %	Computation Energy $\times 10^{-7}$	Communication Energy $\times 10^{-7}$	Total Energy $\times 10^{-7}$
E (without SA)-S		71.16	79.39	75.16	1.18	0	1.18
E (with SA)	60	75.93	80.76	77.95	2.50	0	2.50
E (with SA)	70	76.22	81.26	78.32	3.22	0	3.22
E (with SA)	80	79.15	83.20	80.83	4.41	0	4.41
E (with SA)	90	79.94	83.25	81.31	6.67	0	6.67
E (without SA)-C		80.87	84.88	82.53	13.11	0	13.11
$E \xrightarrow{\textbf{BLE}} F$ **(with SA)**	60	75.93	80.76	77.95	0.79	1.17	1.97
$E \xrightarrow{\textbf{BLE}} F$ **(with SA)**	70	76.22	81.26	78.32	0.79	1.80	2.60
$E \xrightarrow{\textbf{BLE}} F$ **(with SA)**	80	79.15	83.20	80.83	0.78	2.87	3.65
$E \xrightarrow{\textbf{BLE}} F$ **(with SA)**	90	79.94	83.25	81.31	0.78	4.88	5.66
$E \xrightarrow{\textbf{BLE}} F$ **(without SA)**		80.87	84.88	82.53	0.75	10.62	11.37

- In the case that the fog device is not available, either because of poor Bluetooth connection or because the fog device is out of charge, the application is executed on the edge device. In this case, we can decrease the confidence threshold so that the complex classifier is invoked less frequently. This, of course, depends also on the criticality of the situation.
- In our case study, the communication with the cloud engines via LoRa protocol is only used to notify the hospital in case of emergency for rescue, due to the limited bandwidth and the major energy overhead of transmission via this protocol. Thus, we do not use this protocol to transmit the ECG signal to the cloud engines.

Although in our epilepsy detection system, the distribution of tasks over edge and fog provides us with the best trade-off among the performance, latency, and energy, this result cannot be extended to all other biomedical applications. That is, based on the volume of data that is processed (data intensive) and the complexity of computation that is performed by the system (computation intensive), other scenarios can be shown to provide the best outcome among the four possible options.

Let us now consider four possible scenarios. The first scenario belongs to the category of applications that are handled by the edge device. These are applications with a computation load within the edge energy budget, i.e., executing these applications on the edge sensors is more energy/latency efficient than transferring the data to the fog/cloud layer to perform the computation on the fog/cloud layer. On the other hand, the second scenario, in which the tasks are distributed to the fog device, is favorable if transferring the data to the fog layer is more energy-efficient compared to the computation over the edge sensor. Then, we have the third scenario in which the fog layer is used as a gateway to the cloud, which is suitable for the very high-complexity algorithms that can only be executed on the cloud engines. The third and second scenarios are the same in terms of edge-sensor energy efficiency, while the end-to-end latency depends on the computational complexity of tasks to be performed and the volume of data to be transferred to the fog/cloud. Finally, in the fourth scenario, the edge sensor directly communicates with the cloud engines via LoRa protocol, which is limited in terms of bandwidth and has huge energy overheads, hence relevant for informing the hospitals in case of emergencies.

Table III: Summarizing the trade-offs between different modes of system with and without self-aware energy management (values are for 60 seconds of data acquisition and analysis). The energies are normalized with respect to the INYU battery power (710 mAh).

Scenario	Performance %	Latency (ms)	Energy $\times 10^{-7}$
E (without SA)-C	**82.53**	3270.80	13.11
E (without SA)-S	75.16	554.00	1.18
E (with SA)	80.83	**1287.54**	4.41
$E \xrightarrow{\textbf{BLE}} F$	80.83	1420.24	**3.65**
$E \xrightarrow{\textbf{BLE}} F \xrightarrow{\textbf{WiFi}} C$	80.83	2633.12	3.65
$E \xrightarrow{\textbf{LoRa}} C$	80.83	1.06×10^9	2369.84

Let us consider two other applications, which belong to different scenarios than our case study of the epileptic seizure detection system. The first case study is the obstructive sleep apnea monitoring system proposed in [30]. In this application, the system operates more efficiently locally on the edge sensor, i.e., the first discussed scenario, mainly due to the highly optimized and lightweight algorithms adopted.

The second application is training an epileptic seizure detection classifier, using the data from several patients. Note that the training phase of machine-learning algorithms often involves solving complex optimization problems, which is substantially more complicated when compared to the inference phase. In such applications, the computational complexity involved in training a state-of-the-art classifier often is beyond the capacity of the edge and fog devices. Therefore, such complex training algorithms need to be performed on the cloud engines, based on the data transferred to the cloud from the edge sensors.

Finally, considering the very limited bandwidth and huge energy consumption of LoRa/Sigfox, we believe that the case

of direct communication between the edge sensors and the cloud engines is the most useful in reliably notifying the emergency units in case of life-threatening events such as epileptic seizures, as the fog devices (e.g., mobile phone) might be out of reach and unavailable.

VIII. Conclusion

In this article, we have proposed a methodology to distribute the complex and energy consuming machine-learning computations to the fogs/clouds, based on the notion of self-awareness that takes into account the complexity and reliability of the algorithm. We have also analyzed the trade-offs in terms of energy consumption, latency, and performance of different Internet of Things (IoT) solutions. Then, we have considered the epileptic seizure detection problem, as our real-world case study, to demonstrate the importance of our proposed resource-aware distributed health monitoring methodology. Overall, analyzing different scenarios with and without self-awareness shows that using distributed epilepsy monitoring can result in 13-21% energy reduction while self-awareness can reduce the energy by 49-81%. This is while the detection performance is only reduced by 1.22-4.58% making the advantage of distributed health monitoring and self-awareness more than its overhead.

Acknowledgements

This work has been partially supported by the MyPre-Health research project (Hasler Foundation project No. 16073), the ML-Edge research grant by the Swiss NSF (GA No. 200020_182009/1), the EC H2020 DeepHealth project (GA No. 825111), and the Human Brain Project (HBP) SGA2 (GA No. 785907).

References

[1] G. Surrel, A. Aminifar, F. Rincon, S. Murali, and D. Atienza, "Online obstructive sleep apnea detection on wearable devices," in *IEEE Transactions on Biomedical Circuits and Systems (TBioCAS), 2018*. IEEE, 2018.

[2] D. Sopic, A. Aminifar, A. Aminifar, and D. Atienza, "Real-time event-driven classification technique for early detection and prevention of myocardial infarction on wearable systems," *IEEE transactions on biomedical circuits and systems*, vol. 12, no. 5, pp. 982–992, 2018.

[3] H. Jung, "Cisco visual networking index: global mobile data traffic forecast update 2010–2015," Technical report, Cisco Systems Inc, Tech. Rep., 2011.

[4] H. Sun, Z. Zhang, R. Q. Hu, and Y. Qian, "Wearable communications in 5g: challenges and enabling technologies," *ieee vehicular technology magazine*, vol. 13, no. 3, pp. 100–109, 2018.

[5] P. R. Lewis, M. Platzner, B. Rinner, J. Tørresen, and X. Yao, *Self-aware Computing Systems*. Springer, 2016.

[6] P. R. Lewis, A. Chandra, S. Parsons, E. Robinson, K. Glette, R. Bahsoon, J. Torresen, and X. Yao, "A survey of self-awareness and its application in computing systems," in *Self-Adaptive and Self-Organizing Systems Workshops (SASOW), 2011 Fifth IEEE Conference on*. IEEE, 2011, pp. 102–107.

[7] W. H. Organization. (2016) Epilepsy. [Online]. Available: http://www.who.int/mental_health/neurology/epilepsy/en/index.html

[8] D. J. Thurman, D. C. Hesdorffer, and J. A. French, "Sudden unexpected death in epilepsy: assessing the public health burden," *Epilepsia*, vol. 55, no. 10, pp. 1479–1485, 2014.

[9] S. Shorvon and T. Tomson, "Sudden unexpected death in epilepsy," *The Lancet*, vol. 378, no. 9808, pp. 2028–2038, 2011.

[10] A. Van de Vel, K. Cuppens, B. Bonroy, M. Milosevic, K. Jansen, S. Van Huffel, B. Vanrumste, P. Cras, L. Lagae, and B. Ceulemans, "Non-eeg seizure detection systems and potential sudep prevention: State of the art: Review and update," *Seizure*, vol. 41, pp. 141–153, 2016.

[11] C. Hoppe, M. Feldmann, B. Blachut, R. Surges, C. E. Elger, and C. Helmstaedter, "Novel techniques for automated seizure registration: patients' wants and needs," *Epilepsy & Behavior*, vol. 52, pp. 1–7, 2015.

[12] F. Forooghifar, A. Aminifar, L. Cammoun, I. Wisniewski, C. Ciumas, P. Ryvlin, and D. Atienza, "A self-aware epilepsy monitoring system for real-time epileptic seizure detection," in *ACM/Springer Mobile Networks and Applications (MONET)*. ACM/Springer, 2019, pp. 1–14.

[13] F. Forooghifar, A. Aminifar, and D. Atienza, "Self-aware wearable systems in epileptic seizure detection," in *Proceedings of Euromicro Conference on Digital System Design (DSD) 2018*. IEEE, 2018.

[14] S. Murali, F. Rincon, and D. Atienza, "A wearable device for physical and emotional health monitoring," in *Computing in Cardiology Conference (CinC), 2015*. IEEE, 2015, pp. 121–124.

[15] M. Ihle, H. Feldwisch-Drentrup, C. A. Teixeira, A. Witon, B. Schelter, J. Timmer, and A. Schulze-Bonhage, "Epilepsiae–a european epilepsy database," *Computer methods and programs in biomedicine*, vol. 106, no. 3, pp. 127–138, 2012.

[16] M. Hassanalieragh, A. Page, T. Soyata, G. Sharma, M. Aktas, G. Mateos, B. Kantarci, and S. Andreescu, "Health monitoring and management using internet-of-things (iot) sensing with cloud-based processing: Opportunities and challenges," in *2015 IEEE International Conference on Services Computing*. IEEE, 2015, pp. 285–292.

[17] B. Xu, L. Xu, H. Cai, L. Jiang, Y. Luo, and Y. Gu, "The design of an m-health monitoring system based on a cloud computing platform," *Enterprise Information Systems*, vol. 11, no. 1, pp. 17–36, 2017.

[18] M. S. Hossain and G. Muhammad, "Cloud-assisted industrial internet of things (iiot)–enabled framework for health monitoring," *Computer Networks*, vol. 101, pp. 192–202, 2016.

[19] M. Elhoseny, A. Abdelaziz, A. S. Salama, A. M. Riad, K. Muhammad, and A. K. Sangaiah, "A hybrid model of internet of things and cloud computing to manage big data in health services applications," *Future generation computer systems*, vol. 86, pp. 1383–1394, 2018.

[20] T. N. Gia, M. Jiang, A.-M. Rahmani, T. Westerlund, P. Liljeberg, and H. Tenhunen, "Fog computing in healthcare internet of things: A case study on ecg feature extraction," in *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*. IEEE, 2015, pp. 356–363.

[21] A. M. Rahmani, T. N. Gia, B. Negash, A. Anzanpour, I. Azimi, M. Jiang, and P. Liljeberg, "Exploiting smart e-health gateways at the edge of healthcare internet-of-things: A fog computing approach," *Future Generation Computer Systems*, vol. 78, pp. 641–658, 2018.

[22] I. Azimi, J. Takalo-Mattila, A. Anzanpour, A. M. Rahmani, J.-P. Soininen, and P. Liljeberg, "Empowering healthcare iot systems with hierarchical edge-based deep learning," in *2018 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*. IEEE, 2018, pp. 63–68.

[23] A. Page, M. Hassanalieragh, T. Soyata, M. K. Aktas, B. Kantarci, and S. Andreescu, "Conceptualizing a real-time remote cardiac health monitoring system," in *Medical Imaging: Concepts, Methodologies, Tools, and Applications*. IGI Global, 2017, pp. 160–193.

[24] A. Anzanpour, H. Rashid, A. M. Rahmani, A. Jantsch, N. Dutt, and P. Liljeberg, "Energy-efficient and reliable wearable internet-of-things through fog-assisted dynamic goal management," *Procedia Computer Science*, vol. 151, pp. 493–500, 2019.

[25] J. Mohammed, C.-H. Lung, A. Ocneanu, A. Thakral, C. Jones, and A. Adler, "Internet of things: Remote patient monitoring using web services and cloud computing," in *2014 IEEE International Conference on Internet of Things (iThings), and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom)*. IEEE, 2014, pp. 256–263.

[26] J. H. Abawajy and M. M. Hassan, "Federated internet of things and cloud computing pervasive patient health monitoring system," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 48–53, 2017.

[27] H. Kolamunna, J. Chauhan, Y. Hu, K. Thilakarathna, D. Perino, D. Makaroff, and A. Seneviratne, "Are wearable devices ready for https? measuring the cost of secure communication protocols on wearable devices," *arXiv preprint arXiv:1608.04180*, 2016.

[28] C. Ragona, F. Granelli, C. Fiandrino, D. Kliazovich, and P. Bouvry, "Energy-efficient computation offloading for wearable devices and smartphones in mobile cloud computing," in *2015 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2015, pp. 1–6.

[29] S. Shahhosseini, I. Azimi, A. Anzanpour, A. Jantsch, P. Liljeberg, N. Dutt, and A. M. Rahmani, "Dynamic computation migration at the edge: Is there an optimal choice?" in *Great Lakes Symposium on VLSI (GLSVLSI)*, 2019, pp. 519–524.
[30] G. Surrel, T. Teijeiro, M. Chevrier, A. Aminifar, and D. Atienza, "Event-triggered sensing for high-quality and low-power cardiovascular monitoring systems," in *IEEE Design & Test*. IEEE, 2019.
[31] S. Sarma, N. Dutt, P. Gupta, A. Nicolau, and N. Venkatasubramanian, "On-chip self-awareness using cyberphysical-systems-on-chip (cps-soc)," in *Proceedings of the 2014 International Conference on Hardware/Software Codesign and System Synthesis*. ACM, 2014, p. 22.
[32] J. S. Preden, K. Tammemäe, A. Jantsch, M. Leier, A. Riid, and E. Calis, "The benefits of self-awareness and attention in fog and mist computing," *Computer*, vol. 48, no. 7, pp. 37–45, 2015.
[33] N. Dutt, A. Jantsch, and S. Sarma, "Toward smart embedded systems: A self-aware system-on-chip (soc) perspective," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 15, no. 2, p. 22, 2016.
[34] N. Taherinejad, "Ieee life sciences," 2019.
[35] A. Aminifar, *Analysis, design, and optimization of embedded control systems*. Linköping University Electronic Press, 2016.
[36] I. Azimi, A. Anzanpour, A. M. Rahmani, P. Liljeberg, and H. Tenhunen, "Self-aware early warning score system for iot-based personalized healthcare," in *eHealth 360*. Springer, 2017, pp. 49–55.
[37] K. Tammemäe, A. Jantsch, A. Kuusik, J.-S. Preden, and E. Õunapuu, "Self-aware fog computing in private and secure spheres," in *Fog Computing in the Internet of Things*. Springer, 2018, pp. 71–99.
[38] A. Anzanpour, I. Azimi, M. Götzinger, A. M. Rahmani, N. TaheriNejad, P. Liljeberg, A. Jantsch, and N. Dutt, "Self-awareness in remote health monitoring systems using wearable electronics," in *Proceedings of the Conference on Design, Automation & Test in Europe*. European Design and Automation Association, 2017, pp. 1056–1061.
[39] N. TaheriNejad, A. Jantsch, and D. Pollreisz, "Comprehensive observation and its role in self-awareness; an emotion recognition system example," *Self*, vol. 11, p. 1, 2016.
[40] M. Götzinger, N. Taherinejad, A. M. Rahmani, P. Liljeberg, A. Jantsch, and H. Tenhunen, "Enhancing the early warning score system using data confidence," in *International Conference on Wireless Mobile Communication and Healthcare*. Springer, 2016, pp. 91–99.
[41] D. Pascual, A. Aminifar, and D. Atienza, "A self-learning methodology for epileptic seizure detection with minimally-supervised edge labeling," in *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2019, pp. 764–769.
[42] C. Gomez, J. Oller, and J. Paradells, "Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology," *Sensors*, vol. 12, no. 9, pp. 11 734–11 753, 2012.
[43] N. Semiconductor. (2019) Online power profiler. [Online]. Available: https://devzone.nordicsemi.com/power/
[44] S. Support. (2019) Sigfox documentation. [Online]. Available: https://support.sigfox.com/docs
[45] the things network. (2019) Lorawan overview. [Online]. Available: https://www.thethingsnetwork.org/docs/lorawan/
[46] NetSpot. (2019) Wifi standards in a nutshell. [Online]. Available: https://www.netspotapp.com/explaining-wifi-standards.html
[47] A. Pullini, D. Rossi, I. Loi, G. Tagliavini, and L. Benini, "Mr. wolf: An energy-precision scalable parallel ultra low power soc for iot edge processing," *IEEE Journal of Solid-State Circuits*, 2019.
[48] A. Pahlevan, "Multi-objective system-level management of green cloud data centers," p. 181, 2019.
[49] A. Garofalo, M. Rusci, F. Conti, D. Rossi, and L. Benini, "Pulp-nn: Accelerating quantized neural networks on parallel ultra-low-power risc-v processors," *arXiv preprint arXiv:1908.11263*, 2019.
[50] Y. Zhang, Y. Guo, P. Yang, W. Chen, and B. Lo, "Epilepsy seizure prediction on eeg using common spatial pattern and convolutional neural network," *IEEE Journal of Biomedical and Health Informatics*, 2019.
[51] D. Ravì, C. Wong, F. Deligianni, M. Berthelot, J. Andreu-Perez, B. Lo, and G.-Z. Yang, "Deep learning for health informatics," *IEEE journal of biomedical and health informatics*, vol. 21, no. 1, pp. 4–21, 2016.
[52] D. Ravi, C. Wong, B. Lo, and G.-Z. Yang, "A deep learning approach to on-node sensor data analytics for mobile or wearable devices," *IEEE journal of biomedical and health informatics*, vol. 21, no. 1, pp. 56–64, 2016.
[53] A. Jafari, A. Ganesan, C. S. K. Thalisetty, V. Sivasubramanian, T. Oates, and T. Mohsenin, "Sensornet: A scalable and low-power deep convolutional neural network for multimodal data classification," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 1, pp. 274–287, 2018.
[54] D. Biswas, L. Everson, M. Liu, M. Panwar, B.-E. Verhoef, S. Patki, C. H. Kim, A. Acharyya, C. Van Hoof, M. Konijnenburg *et al.*, "Cornet: Deep learning framework for ppg-based heart rate estimation and biometric identification in ambulant environment," *IEEE transactions on biomedical circuits and systems*, vol. 13, no. 2, pp. 282–291, 2019.
[55] H. Li, K. Ota, and M. Dong, "Learning iot in edge: Deep learning for the internet of things with edge computing," *IEEE Network*, vol. 32, no. 1, pp. 96–101, 2018.
[56] M. Qaraqe, M. Ismail, E. Serpedin, and H. Zulfi, "Epileptic seizure onset detection based on eeg and ecg data fusion," *Epilepsy & Behavior*, vol. 58, pp. 48–60, 2016.
[57] P. J. Fleming and J. J. Wallace, "How not to lie with statistics: the correct way to summarize benchmark results," *Communications of the ACM*, vol. 29, no. 3, pp. 218–221, 1986.
[58] *STM32L151RD*, Ultra-low-power ARM Cortex-M3 MCU with 384 Kbytes Flash, 32 MHz CPU, USB, 3xOp-amp - STMicroelectronics.
[59] *ECG electrode for sensitive skin*, Ambu BlueSensor VLC.
[60] T. Instruments. (2018) Ads7142. [Online]. Available: http://www.ti.com/product/ADS7142/description

Farnaz Forooghifar is Ph.D. researcher in the Embedded Systems Laboratory (ESL) at the Swiss Federal Institute of Technology Lausanne (EPFL), Switzerland. She received her B.Sc. and M.Sc. in electrical engineering from University of Tehran, Iran, in 2014 and 2017. Her research interests include real-time health monitoring systems, embedded systems design, approximate computing and parallel processing.

Amir Aminifar received his Ph.D. from the Swedish National Computer Science Graduate School (CUGS), Linköping University, Sweden, in 2016. During 2014-2015, he visited the Cyber-Physical Systems Laboratory of the University of California, Los Angeles (UCLA), USA and the Real-Time Systems Laboratory of Scuola Superiore Sant'Anna, Italy. He is currently a research scientist at the Embedded Systems Laboratory of the Swiss Federal Institute of Technology Lausanne (EPFL), Switzerland. His current research interests are centered around mobile health technologies and medical informatics for reliable detection and prediction of pathological health conditions.

David Atienza (M'05-SM'13-F'16) is associate professor of electrical and computer engineering, and director of the Embedded Systems Laboratory (ESL) at the Swiss Federal Institute of Technology Lausanne (EPFL), Switzerland. He received his Ph.D. in computer science and engineering from UCM, Spain, and IMEC, Belgium, in 2005. His research interests include system-level design and thermal-aware optimization methodologies for 2D/3D high-performance multi-processor system-on-chip (MPSoC) and ultra-low power system architectures for wireless body sensor nodes. He is a co-author of more than 250 papers in peer-reviewed international journals and conferences, several book chapters, and seven patents. Dr. Atienza received the DAC Under-40 Innovators Award in 2018, IEEE TCCPS Mid-Career Award in 2018, an ERC Consolidator Grant in 2016, the IEEE CEDA Early Career Award in 2013, the ACM SIGDA Outstanding New Faculty Award in 2012, and a Faculty Award from Sun Labs at Oracle in 2011. He served as DATE 2015 Program Chair and DATE 2017 General Chair. He is an ACM Distinguished Member and an IEEE Fellow.