Toward Ultra-Low-Power Remote Health Monitoring: An Optimal and Adaptive Compressed Sensing Framework for Activity Recognition

Josué Pagán, *Student Member, IEEE*, Ramin Fallahzadeh, *Student Member, IEEE*, Mahdi Pedram, *Student Member, IEEE*, José L. Risco-Martín, José M. Moya, José L. Ayala, *Senior Member, IEEE*, and Hassan Ghasemzadeh, *Senior Member, IEEE*

Abstract—Activity recognition, as an important component of behavioral monitoring and intervention, has attracted enormous attention, especially in Mobile Cloud Computing (MCC) and Remote Health Monitoring (RHM) paradigms. While recently resource constrained wearable devices have been gaining popularity, their battery life is limited and constrained by the frequent wireless transmission of data to more computationally powerful back-ends. This paper proposes an ultra-low power activity recognition system using a novel adaptive compressed sensing technique that aims to minimize transmission costs. Coarse-grained on-body sensor localization and unsupervised clustering modules are devised to autonomously reconfigure the compressed sensing module for further power saving. We perform a thorough heuristic optimization using Grammatical Evolution (GE) to ensure minimal computation overhead of the proposed methodology. Our evaluation on a real-world dataset and a low power wearable sensing node demonstrates that our approach can reduce the energy consumption of the wireless data transmission up to 81.2 and 61.5 percent, with up to 60.6 and 35.0 percent overall power savings in comparison with baseline and a naive state-of-the-art approaches, respectively. These solutions lead to an average activity recognition accuracy of 89.0 percent—only 4.8 percent less than the baseline accuracy—while having a negligible energy overhead of on-node computation.

Index Terms—Compressed sensing, activity recognition, feature selection, energy efficiency, ultra-low power, optimization, adaptive

1 Introduction

The Internet of Things (IoT) brings new opportunities for health care in unobtrusive monitoring scenarios (eHealth), for proactive personal eHealth control [1], sports training applications [2], health-related timely interventions [3], *etc*. Among all these applications, the core functionality is activity recognition, which is a key enabler in context aware systems. Activity recognition has numerous health-related applications such as prevention, diagnosis, and monitoring of several illnesses related to functional impairment [4] or Parkinson disease [5]. In addition, it plays a crucial role in in Ambient Assisted Living tools designed for elders [6].

Dozens of private companies have reached the market of wearables and activity recognition. Smartwatches like Apple Watch, sport chest-bands or wristbands such as Microsoft Band, are just some examples. New technologies in wearable devices provide more efficient processor units to compute even more complex activity recognition algorithms. The Snapdragon 400 processor is an example of a high performance processor used in many of state-of-the-art smartwatches. Low performance microcontrollers (MCUs) can be found in many different wearable devices, as the 16-bit MSP430 MCU—that controls the open-source hardware platform Nike Fuelband—or the 8-bit MCUs of Sillicon-Labs for wearables.

Many current wearable activity recognition technologies require that end-users follow certain protocols while being monitored using sensors that are coupled with the human body. For example, they must carry a wearable node at certain body location all the time; otherwise, physical activity measurements (e.g., type and intensity of activities/movements) will be inaccurate [7]. One study [8], showed that in absence of automatic sensor localization, the accuracy of an activity recognition algorithm drops to 33.6 percent from 98.4 percent. This means the signal attributes of wearables are tightly coupled with their associated on-body node location. Further more limiting the wearable on-body location,

- *J. Pagán, J.L. Risco-Martín, and J.L. Ayala are with the Department of Computer Architecture and Automation, Universidad Complutense de Madrid, Madrid 28040, Spain, and the Center for Computational Simulation, Universidad Politécnica de Madrid, Campus de Montegancedo, Boadilla del Monte, Madrid 28660, Spain. E-mail: {jpagan, jlrisco, jayala}@ucm.es.*
- *R. Fallahzadeh, M. Pedram, and H. Ghasemzadeh are with the School of Electrical Engineering and Computer Science, Washington State University, Pullman, WA 99164. E-mail: {rfallahz, mahdi.pedram, hassan}@eecs.wsu.edu.*
- *J.M. Moya is with the Universidad Politécnica de Madrid, Madrid 28040, Spain, and the Center for Computational Simulation, Universidad Politécnica de Madrid, Campus de Montegancedo, Boadilla del Monte, Madrid 28660, Spain. E-mail: josem@die.upm.es.*

Manuscript received 27 Sept. 2017; revised 2 May 2018; accepted 28 May 2018. Date of publication 4 June 2018; date of current version 4 Feb. 2019. (Corresponding author: Josué Pagán.)
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TMC.2018.2843373

imposes user discomfort and limits the performance capacity of wearables. This issue has many real-world implications. For example, while one person may prefer to use his/her smartphone in his pocket, another person may prefer to carry the smartphone in a purse or a belt-clip. While one person may prefer to use a wrist-band sensor on his right wrist, another person may prefer a left wrist setting. A new trend is wearables allows users to wear wearables in more than one predefined on-body location. The emerging clip-on fitness trackers such as 'Misfit Shine 2' and 'Fitbit Zip' are examples of this new trend.

Using the Snapdragon 400 processor, Bhattacharya et al. [9] present a methodology for activity recognition using deep learning. These algorithms need computational resources that might not be present in more simple devices. The authors claimed that their system could be used and implemented on off-the-shelf devices, however, they did not provide experimental proof of low power consumption. Maurer et al. developed a framework called *eWatch* [10], which uses several sensors for location and activity recognition but their main goal was not on power efficiency. Therefore, it is unlikely that their system can be utilized in every day settings because of limited battery lifetime. In [11], Patterson et al. described a context-aware graphical model to perform a fine-grained activity recognition using Hidden Markov Models. The system considers a different sensor modality, which uses tags in objects and a RFID antenna. While the methodology is interesting, its application is limited to smart environments such as smart homes or smart nursing facilities. The authors in [12], presented a Mobile Sensing Platform for daily activity recognition. The system uses a large number of sensors, which results in reduced battery life. Thus, due to the aforementioned limitations, we will use low performance MCUs to deal with the activity recognition problem, because they are cheap, highly integrated, and low power.

Physical activity recognition, as one of the most popular applications of human-centered IoT, is the specific target of this study. However, this methodology can be potentially extended to other applications of IoT where power efficiency is an obstacle, such as most of eHealth applications involving continuous monitoring. For example, monitoring Heart Rate Variability (HRV) for detection of cardiac failures, or photoplethysmography (PPG) for information extraction such as blood pressure or oxygen saturation.

In this paradigm we can distinguish, at least, two scenarios: i) a sensing node with some knowledge (i.e., simple feature extractions) that transmits data to a gateway platform (e.g., a smartphone) in charge of the activity recognition; or ii) scenario composed of a high performance device (e.g., a smartphone) that monitors and performs light to moderate computing tasks (i.e., activity recognition) but transmits the data to a back-end server, as a data center, for storage or high-demanding computation. Many commercial smartphone applications running activity recognition tasks compute these in the server side [13]—which makes this task independent of the wearable device but, on the contrary, demands frequent wireless data transmission. Offloading data from nodes to Cloud servers remains into the Mobile Cloud Computing (MCC) [14] and Remote Health Monitoring (RHM) paradigms, leading to high energy and economic savings [15]. In this paper we focus on the *scenario i* (one *Plug&Play* sensing node and a smartphone as back-end). However, the methodology developed also fits in the *scenario ii*.

Developing energy-efficient system-level solutions and computationally simple embedded software are warranted in order to drive the wearable technology forward. At the hardware system level there are ingenious solutions in the field of energy harvesting from kinetic energy using piezoelectric [16] and capacitive [17] technologies that reduce considerably the power consumption of the sensing nodes. They have studied this for application in activity recognition problems, however these solutions are still far from commercialization. In this paper, we devise an ultra-low-power solution applicable to state-of-the-art sensing nodes based on adaptive Compressed Sensing (CS) methodology to reduce the amount of data being transmitted. Compressed sensing allows us to create a representation of the original data in a transformed domain reducing the amount of data to transmit with an acceptable minimal information loss. Therefore, compressed sensing is more applicable when having a periodic signal such as ECG [18], [19], EEG [20], PPG [21] or motion data.

Our work presents a low-power optimized temporal adaptive compressed sensing framework for activity recognition. A preliminary version of this study has been published previously in [22]. To do this, a coarse-grained activity recognition is performed on-node. By motion data, the system i) implements coarse-grained node localization and ii) classifies the *signal type* to iii) automatically update the compressed sensing rate in order to rigorously reduce the amount of data being transmitted. We further implement our platform on a real device where the energy consumption has been measured. Our results from real world experiments show that the overhead of the methodology is negligible compared to the overall energy saving achieved in wireless transmission. Eventually, we compare the achieved savings to the state-of-the-art compressed sensing approaches and the baseline.

The remainder of this paper is as follows. In Section 2, a general overview of the proposed methodology is described. Basic concepts about algorithms and techniques applied are then introduce in Section 3. In Section 4, the main modules of the system are detailed. Section 5 shows the evaluation and the discussion of the results. In Section 6 the authors discuss about the paradigm of real-time dependability in real-world applications. Finally, conclusions of this work are drawn in Section 7.

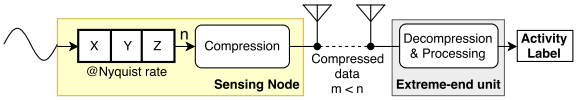
2 Methodology Overview

This section describes the high level overview of the proposed system. Furthermore, we compare our framework to the traditional compression techniques and the state-of-the-art techniques using compressed sensing approaches.

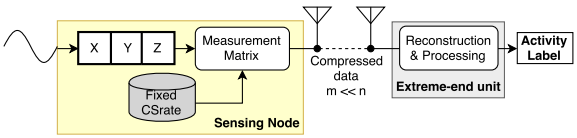
An activity recognition framework consists of two differentiated parts commonly described in most IoT systems: the sensing node and the back-end computing unit. Our scheme consists of a resource constrained and low cost activity tracking device and a higher performance back-end, e.g., a smartphone. However, this methodology can also be applied when the activity tracker is a smartphone and the back-end is a server hosted in a data center. In the following, we compare different compression strategies in sensing nodes, while the back-end unit remains practically unmodified (as shown in Fig. 1c).

2.1 State-of-the-Art Methodologies

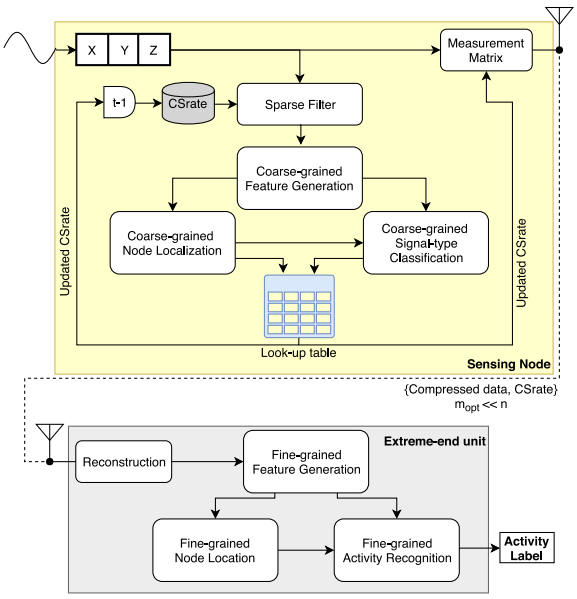
The basic approach (Fig. 1a) as opposed to compressed sensing, uses conventional compression/decompression



(a) Basic compression scheme.



(b) Naive compressed sensing activity recognition system in the state-of-the-art.



(c) Adaptive temporal compressed sensing activity recognition system.

Fig. 1. Overview of the basic, the state-of-the-art (naive), and the proposed compression methodologies. The back-end unit is detailed only in the last one.

algorithms (such as ZIP compression). Sensor reading are compressed locally, and are decompressed and fed into a fine-grained activity recognition module after transmission to a back-end station. The superiority of compressed sensing over such conventional algorithms, in motion data application, has been shown in literature [13] both in terms of computation complexity and compression performance.

The state-of-the-art approach (Fig. 1b) uses optimized compressed sensing technique, but lacks any intelligent on-node adaptivity. In other words, sensor readings are optimized using a fixed compression rate (learned and optimized off-line). While it can be conceived as a considerable improvement over the basic approach, its minimization is limited because of the absence of sensitivity to its context (e.g., varied attributes of the input data, on-body sensor location). The performance of the framework proposed in this paper is compared with the aforementioned framework in Fig. 1b, and also with a baseline approach where data is transmitted without compression.

In order to eliminate the real-time feedback dependency and improve the energy efficiency of the activity recognition system, we propose a temporal adaptive compressed sensing framework (Fig. 1c).

2.2 Proposed Optimized Methodology

The proposed adaptive compressed sensing activity recognition system (Fig. 1c), inherits the capabilities of sate-of-the-art approach and introduces novel low-cost coarse-grained input type detection and on-body node localization algorithms. The context-awareness provided by these two modules enables the node to autonomously detect such changes and adjust the ratio and therefore provide adaptive and robust optimal compression.

The sensing node can be potentially worn anywhere on the body: pants pockets, chest, arm, and wrists are among the most common on-body node location in activity recognition wearables (for simplicity throughout the paper we refer to *on-body node location* as *location*). In our case, the sensing node performs a novel dynamic compressed sensing scheme to reduce the amount of data sensed transmitted and thus enhance the battery life. The computation server (i.e., back-end) receives the data and reconstructs an approximation of the original signal to apply activity recognition techniques after that on the application side.

Compressed sensing technique can reduce the amount of data being sensed, processed, and transmitted with minimal impact on the performance of the accuracy of the system (see Section 3.1). This technique can be applied at i) at sampling rate [23] (adaptive hardware sampling), or ii) once the data are acquired at an adequate sampling rate.

The sensing node utilizes the sensor readings to extract features for continuous context learning. Every time a data segment is gathered and processed, the sensing node, if needed, re-configures the compression ratio. Data segments are values of several seconds of a triaxial accelerometer. Features will be extracted from the compressed data. Using these features two tasks are performed: i) coarse-grained node localization and ii) coarse-grained and location dependent signal type detection as shown in Fig. 1c.

We note that we make a distinction between the activity label and the signal type. A activity label (i.e., physical activity label) is any activity or movement that the subject wearing the device performs, such as: walk or jump. On the other hand, a signal type is an abstract and undetermined concept that refers to the idea that, signals having different morphology and origin, can lead to an identical data processing regardless of their corresponding activity labels. Thus, many different activities can share the same signal type and therefore the same compression ratio can be applied. From the perspective of a sensing node worn on an arm, there is little to no difference between the two activities with respect to compression ratio. Using the signal type instead of simply relying on the activity label enables us to leverage adaptivity in an unsupervised fashion, meaning we do not need to label the training/test data to learn the signal type.

Under this circumstance, two optimization problems can be defined: i) detecting the optimal number of signal types (clusters of signals per on-body location), and ii) assigning a compression ratio to each signal cluster such that the energy consumption is minimized. For each location and signal type, a specific compression ratio will be applied to the raw data according to the look-up table (that is trained offline) stored in the local memory of the sensing node as shown in Fig. 1c. The input of this look-up table is the context (i.e., signal type per location) learned by the course-grained module and the output is the context optimized compression ratio. The compressed data and the updated ratio are sent to the back-end unit, where the compressed raw data are recovered. In the back-end, features are extracted again to compute: i) fine-grained node localization, and ii) fine-grained coarse-grained location-aware activity recognition.

In Section 3, we briefly elaborate i) how the compressed sensing technique is performed; ii) how to devise an offline GE-based optimization algorithm; and finally iii) how to perform the activity recognition task. All the modules in Fig. 1c are further elaborated in Section 4.

3 Preliminaries

3.1 Compressed Sensing

Compressed sensing theory states that a signal x can be recovered with minimal information loss by sub-sampling the signal with order $m << n$ where n denotes the Nyquist sampling frequency under sparsity and incoherence assumption [24], [25].

Compressed sensing—as many other compression techniques, such as Wavelet or Fourier Transform—are efficiently applied in periodic signals. Representation of periodic signals in terms of these transformations lead to sparse signals; meaning that the most of the coefficients c will be equal to zero after the transformation. In general, it can be asserted that the movement signals used in this work can be considered periodic signals. We briefly describe how compressed sensing is formulated and applied in our study.

Given x sampled at Nyquist rate n, the signal x can be written as a linear combination as

$$x = \Psi c, \tag{1}$$

where c is the set of sparse coefficients of the signal x after domain transformation given the transformation basis Ψ. In our case, x is a one dimensional sensor reading segment of length p, $\Psi \in \mathbb{R}^{p \times p}$ is a Discrete Cosine Transformation (DCT) matrix. Similarly, the length of c is equal to p.

To represent the sub-sampled signal y, the matrix Φ is used. Φ is the *sensing matrix* and takes q random samples of the signal x, thus

$$y = \Phi x. \tag{2}$$

In our case, signal y is the data being sent from the monitoring device to the back-end unit, and its length is $q \times 1$; and $\Phi \in \mathbb{R}^{q \times p}$. Notice that $q/p = m/n$, and it is referred to as the compression ratio cr. Randomization affects only mildly to the calculus of features because these are based on statistics of the morphology of the signal and not on temporal properties (see Section 5.1).

To recover an approximation x' of the original signal x in the back-end, the estimation of coefficients c is required. Using Eqs. (1) and (2), we can also write the compressed signal y as

$$y = \Phi\Psi c = Ac, \tag{3}$$

where $A \in \mathbb{R}^{q \times p}$ is the *measurement matrix*, defined in Eq. (4). A is the extraction of q rows of the DCT matrix Ψ, and it is known by the sensing node and the receiver-end to compute the estimation c' of coefficients c. Thus, the reconstructed signal $x' = \Psi c' \approx x$

$$A = \Phi\Psi. \tag{4}$$

Computation of coefficients of c' is expensive and has to take place in back-end, with higher performance than the monitoring devices. According to the authors that drove the compressed sensing techniques to its current status [26], [27], each column of matrix A implies to solve a linear system. This is an undetermined linear system and is hard to solve [28]. Therefore, the problem needs to be relaxed to $l_1 - norm$ minimization

$$Minimize \ \ \|c'\|. \tag{5}$$

Subject to

$$\|y - Ac'\| < \epsilon, \tag{6}$$

where ϵ is the margin of reconstruction error. c' can be found in polynomial time using linear programming.

In this work we propose a temporal adaptive compressed sensing approach for activity recognition. To the best of our knowledge, only one work by Yuan et al. [29] presents a time adaptive compressed sensing approach (applied to a video problem). Chiu et al. [30] pose adaption of basis functions of the sensing matrices and adaption of the compression ratio to the problem (energy auditing networks for different services); however, the adaptation is not dynamic.

Compressed sensing is a very well known methodology in the field of image processing. Hardware compressed sensing [31] has been commercialized recently. Unfortunately, there are no hardware implementation in microcontrollers used in embedded wireless monitoring devices, and only a RISC architecture of a specific-application processor for compressed sampling is described by Constantin et al. in [32]. As a result, in this paper, compressed sensing is implemented as firmware (FW) of the monitoring node.

3.2 Application in Activity Recognition

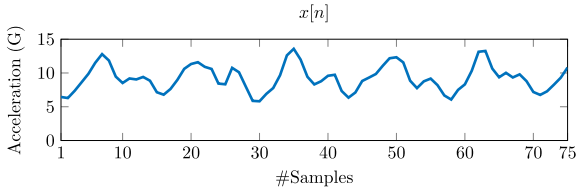
Figs. 2a and 2c are examples of the processing of the compressed sensing algorithm. Fig. 2a represents data from one axis of the accelerometer acquired at $25Hz$. In Fig. 2b the sparse DCT is shown. This signal is sparse enough to be transmitted—decreasing the amount of data and resulting in significant power savings. Randomized sampling of 33 percent of this signal is used to recover x' in Fig. 2c, as an approximation of the original data x. It can be observed that the morphology of the reconstruction is still distinguishable.

As Yang et al. state in [33], the sparsity of the physical activities varies for different types of movements and locations. The authors in [33], evaluate different compression ratios for different activities; they do not evaluate the accuracy over the activity recognition but over the signal reconstruction, and they conclude that this can lead to similar information loss when measured by the Normalized Root-Mean-Square Error (NRMSE). However a supervised classifier (trained with a large enough labeled data) is required to detect changes in activities. We hypothesize that optimization of the compression ratio based on signal-type recognition may result in even more signification energy savings while using an unsupervised approach.

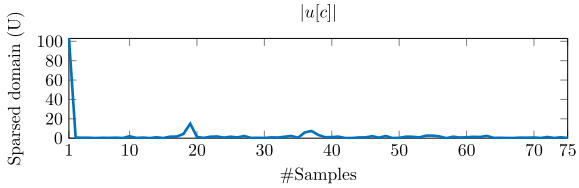
3.3 Optimization Using Grammatical Evolution

As mentioned in Section 2.2, the system running on the sensing node needs to detect the signal type—based on the detected location—in order to apply the optimum cr.

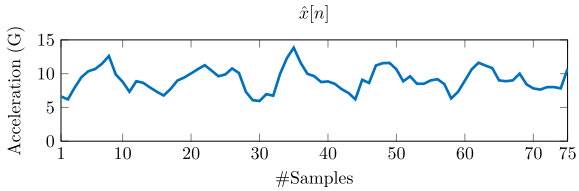
In our framework, it is not necessary to know the activity labels or any other labeled data to work. It is important to highlight that we do unsupervised classification (clustering) and it is one of the strengths of the proposed model as opposed to the labor-intensive supervised classification whose performance is bounded by the provided training data. Instead, this model uses unlabeled data which is abundant and easy to gather. First, we need to optimize our



(a) Three seconds of walking data.



(b) Sparse DCT representation.



(c) Reconstructed signal.

Fig. 2. Sparse recovery of activity *walking* using $cr = 33\%$ and $NRMSE = 0.08$.

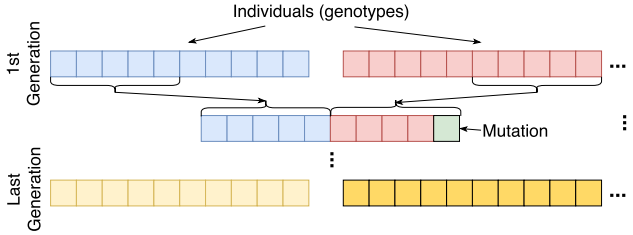
clustering and then we can find the minimum ratio per each discovered cluster. The details of this bi-objective optimization problem are given by the following problems:

Problem 1 (Optimal signal-type clustering per on-body location). *For all the different activities from each on-body node location, find the clustering solution that optimally identifies the types of signals. See details in Section 4.1.4.*

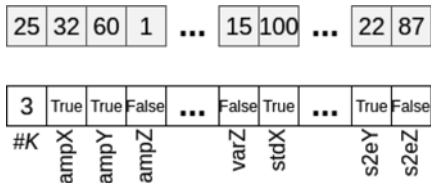
Problem 2 (Optimal compression ratio per cluster). *For each cluster containing a particular signal type per on-body location, find the optimal compression ratio that minimizes the energy consumption of the node and maximizes the accuracy of the activity recognition in the receiver-end.*

In order to tackle these problems, we use GE [34]. GE is a grammar-based form of Genetic Programming (GP), used to generate programs in any language. GE's internal behavior is based on a Genetic Algorithm (GA). GAs are stochastic methods to solve complex optimization problems. Its heuristic search allows to tackle faster big combinational problems, as ours, the optimal solution or really close to this. GE has genotypes—represented as integer numbers—that select production rules from a group expressed in a Backus Naur Form (BNF) and lead to phenotypes. A phenotype is a tree-shape structure which is evaluated in an iterative process. In our case, a phenotype represents the set of activity-features to be selected.

Following with the biological simile, a GA evolves a population formed by a set of individuals (the genotypes) as shown in Fig. 3a. Individuals mutate and mix with each other to create new ones in every generation. Each individual is evaluated, and those of each generation that better fit the objective will survive and evolve with a higher probability in future generations. For an example of a complete process of decodification of a genotype refer to [35]. The gray array in the top of Fig. 3b shows how a genotype looks like. After applying the BNF's rules, in our case, a phenotype looks like the white array shown in the bottom of Fig. 3b.



(a) Example of population. Mix and mutation of individuals are shown.



(b) Chromosome array and its decoded final expression.

Fig. 3. Example of the population and chromosome decodification of the GE problem.

The designed grammar is shown in BNF format in Fig. 5. This grammar allows the optimization of the number of clusters for the signal type detection.

A BNF grammar is represented by a set of parameters in the form $\{N, T, P, S\}$, where N denotes the set of non-terminals (coded symbols), T denotes the set of terminals (decoded expressions), P denotes the set of production rules to substitute the elements of N into T, and S is a non-terminal element of N used as starting symbol. A model (labeled as `Model`) is an array combination of (terminals, T): the number of clusters (`K`), and a set of `F` Boolean values (`Boolean`) that indicate which activity-features $\in$ `F` have to be selected.

Because our approach is based on a multi-objective optimization, we use a multi-objective genetic algorithm inside GE: the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [36]. The main difference between a simple GA and NSGA-II is that at the end, as opposed to a single best solution, a set of non-dominated solutions is obtained (traditionally named as Pareto front). NSGA-II is an elitist approach, what means that a small part of the best candidates remain unchanged into the next generation—they remain as parents of the next generation, which enhance convergence and allows computation of an approximation of the entire Pareto front. Fig. 4 illustrates an example of a two-dimensional Pareto front. Black points in Fig. 4 are the last individuals or solutions of the NSGA-II execution. Red circles surround those solutions that lead to the Pareto front. Refer to [37] for details on multi-objective GE implementation.

3.4 Optimizing the Clustering Scheme

As mentioned in Section 3.3, the optimization problem #1 finds the optimum number of clusters that matches the different number of signal types. To perform this task, a *k-means++* clustering is utilized (see Section 4.1.4). *k-means++* is an unsupervised approach that needs an internal evaluation index to rank each individual (solution) of the GE.

In order to evaluate each individual solution of the GE, an internal evaluation index is needed. There are several internal indices for evaluating a cluster, such as: the Davies-Bouldin index, Dunn index or silhouette coefficient. While any of these indexes could be used in our optimization, the

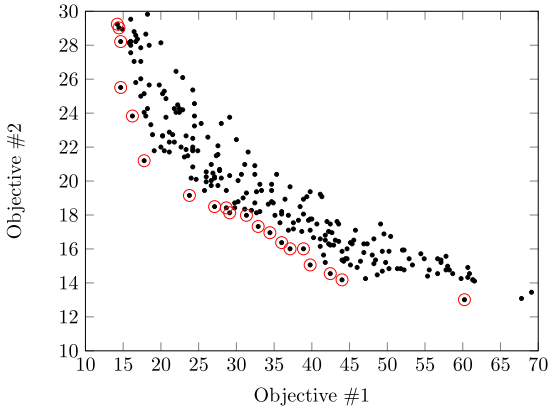


Fig. 4. Example of Pareto front to minimize a two-dimensional objective problem.

former has been employed because of its lower complexity. The Davies-Bouldin index [38] is given by

$$\overline{R} \equiv \frac{1}{K}\sum_{i=1}^{K} \max_{i \neq j}\left(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)}\right), \tag{7}$$

where the parameter K denotes the total number of clusters, the parameter c_i denotes the centroid of cluster i, the parameter σ_i represents the average distance of all elements in cluster i to the centroid c_i, and the parameter $d(c_i, c_j)$ is the distance between centroids c_i and c_j. We use the euclidean distance as the distance metric.

The parameter $\overline{R}$ is non-negative. Lower values of $\overline{R}$ indicate better clustering. $\overline{R}$ relates the scatter of the cluster with the distance between clusters. As the distance between clusters is in the denominator in Eq. (7), $\overline{R}$ is low when clusters are not spread and each one is compact (i.e., the distance between clusters increases).

4 System Detail

In this section, we elaborate each module in Fig. 1c from the perspective of the two functional components of the system: the node, and the back-end computing unit.

The optimization problems have been solved using an implementation of GE and NSGA-II algorithms available in the HERO Java Library [39]. Compressed sensing techniques, *k-means++* clustering, and clustering evaluation have been computed in Matlab. The decompression has been carried out through the method to recover sparse signals via convex programming implemented in Matlab by Romberg and developed in [40]. The activity recognition and the node localization have been computed using WEKA [41].

4.1 Sensing Node View

Sensing nodes acquire data from triaxial accelerometers and store them into segments (s) of few-second sensor data. Nodes can be placed on different locations of the body. For the sake of simplicity, five known locations are considered. For the compressed sensing approach, each data segment s—at time k—is sub-sampled at rate $cr[t-1]$, leading to a new segment s'. This takes place in the *Sparse Filter* module (see Fig. 1c). From this segment, different features are extracted in the *Coarse-grained Feature Generation* block. With these features, the location is determined in the *Coarse-grained node Localization* module. For each location the signal type is detected by the *Coarse-grained Signal Type* module.

```
N = { <Model>, <K>, <Boolean> }
T = { TRUE, FALSE, 2, 3,..., 25 }
S = <Model>

P = {
  I    <Model>     ::= [<K>, <Boolean>, <Boolean>,...,
                       <Boolean>] # Till the total number
                            of features F
  II   <K>         ::= 2|3|4...|25
  III  <Boolean>   ::= TRUE|FALSE
}
```

Fig. 5. BNF grammar used for to solve the problem of optimal signal type classication.

With the location and the signal type, the compression ratio is updated to $cr[t]$ according to the embedded *Look-up table*. The measurement matrix A from Eq. (4) is applied in the *DCT* module and compressed data are transmitted.

4.1.1 Sparse Filter

In this module, the sub-sampled s' sequence is created picking $cr[t-1]$ random samples of segment s. At this point, the compression ratio has not been updated yet. Therefore, transitory states that lead to the selection of non-optimal and delayed compression ratios, may occur between transition of activities. However, these transitions are sparse in time and are insignificant in practice.

4.1.2 Feature Generation

Using the sub-sampled sequence s', 30 morphological features are computed, as it will be shown in Section 5.1. Computation of these features is computationally light and a relatively simple task to perform for resource constrained monitoring nodes.

4.1.3 Coarse-Grained Node Localization

Different data mining algorithms have been trained to classify and detect the node localization: classification trees, support vector machines or *k-means* among others. *Random Forest* [42] has shown the highest accuracy using 10 cross-fold validation. *Random Forest* is a supervised algorithm that generates a set of different trees under certain rules. To classify a new instance—vector of features extracted from sequence s'—the generated trees vote for the most popular class. All classes (locations) must be in the training set.

Random Forest algorithm is a sequence of *if-else* statements which makes it more practical to implement in sensing nodes with constrained computation capacity. The *Random Forest* algorithm is trained offline using all 30 features extracted from uncompressed raw data. In real time, as the features are extracted from sub-sampled data, a coarse-grained node localization will take place.

4.1.4 Coarse-Grained Signal Type Classification

Several known daily and sports activities are considered for this research (see Section 5.1). As defined in Section 2.2, we will classify our movement data into signal types. As was previously mentioned, the cardinality of the set of signal types is unknown (signal type detection is an unsupervised classification problem). To achieve the best performance-power saving trade-off, we need to solve the optimization Problems 1 and 2 stated in Section 3.3:

Optimization Problem #1. Clustering of the optimum number of signal types maximizing the number of clusters $\#k \in K$, the quality Q_{idx} of the final cluster, and minimizing the number of required features $\#f \in F$.

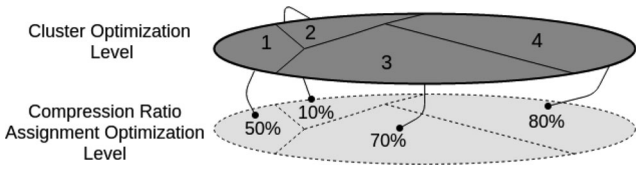


Fig. 6. Per each location: Division of the different signal types found and the assignment of the compression ratios.

As granularity criterion, we consider $K = 2, 3, \ldots, 25$, and $\#k > 1$ as a constraint. This granularity has been established in order to minimize the computation time of the optimization problem. The limit is defined by the upper-bound on the number of clusters possible: 125 ($\Delta_{min}cr = 100 * \frac{1}{125} = 0.8\%$), equal to the size of input data. Maximizing the number of clusters will allow us to apply several compression ratios that lead to a more accurate control of the energy consumption in the sensing node. However, high granularity could lead to close clusters with slightly different compression ratios. To avoid this, we have established a trade-off between the number of clusters and granularity, and thus, we studied only 25 different values, so that the minimum increment of $\Delta_{min}cr = 100 * \frac{25}{125} = 4.0\%$, which is not too low or not too high (for 25 Hz and 5 seconds data length it leads to compression of 5 samples). The quality of the cluster is computed using the Davies-Bouldin index, thus $Q_{idx} = \overline{R}$. $\#f$ will be minimized to reduce the computation in the node, with $F = 1, 2, \ldots, 30$. Given that, the multi-objective problem can be formulated as

$$\min\left(\#f, Q_{idx}, \frac{1}{\#k}\right). \tag{8}$$

Given a number of clusters $\#k \in K$ and a set of features $f' \in F$, the system computes the goodness of the clustering Q_{idx}. Because there are no labels to indicate the different signal types, this is an unsupervised clustering that has been implemented using a *k-means++* algorithm.

Fig. 6 illustrates two optimization levels. The upper one is an example of the optimal partitioning of the data space. In this example, four different types of clusters and the corresponding values are found.

In this optimization, the selected solutions are named as $S^*_{P_{ST}}$. For each one of the selected solutions a compression ratio set is assigned. These appear in the lower abstract plane in Fig. 6, and these values are found through a second optimization problem stated in Section 3.3.

Optimization Problem #2. To find the optimum set of compression ratios $cr_{set,f'}$ for a given set of features f' that maximizes the weighted mean of the compression ratio $\overline{cr}_{set,f'}$ and maximizes the accuracy of the activity recognition α_{AR} in the back-end. Thus, the multi-objective problem states that

$$\min(1 - \overline{cr}_{set}, 1 - \alpha_{AR}). \tag{9}$$

The weighted mean compression ratio is given by

$$\overline{cr}_{set,f'} = \frac{1}{N}\sum_{i=1}^{\#k} n_i cr_{i,f'}, \tag{10}$$

where N is the number of instances in the training set, n_i is the number of instances that have been classified into the cluster i, and $cr_{i,f'}$ is the compression ratio for cluster i using the set of features f'. This optimization—based on each solution $S^*_{P_{ST}}$—leads to a new Pareto front. In this Pareto front, the selected solutions $S^*_{P_{CS}}$ must satisfy

$$\max_{S^*_{P_{CS}}} (\overline{cr}_{set,f'} \mid \varepsilon_{AR} \leq \varepsilon_{th}), \tag{11}$$

to be implemented in the nodes.

Eq. (11) establishes that the solutions to be implemented are the ones with higher weighted mean compression ratio from those solutions which have an error in the activity recognition ε_{AR} less than a threshold ε_{th}. The threshold ε_{th} is a quality criterion set defined as the extra error added to the lower bound error in activity recognition $\varepsilon_{AR} = 100 - \alpha_{AR_{Base.}}$. Here, the extra error has been set to 5 percent. Thus, $\varepsilon_{th} = \varepsilon_{AR} + 5\%$. This value has been chosen based on criterion and experience of the researchers. It has been considered that this value helps to relax conditions to achieve better consumption metrics, as well as to keep a considerable quality level.

The maximum number of clusters established in the previous optimization problem also sets the granularity in the compression ratios. Thus, in the $cr_{set,f'}$ each value $cr_i \in CS = 0, 4, 8, \ldots, 96$—$K = 25$ equidistant levels in the interval [0, 100).

As in the example, incoming data classified as signal type 1 in Fig. 6, is compressed with $cr = 50\%$, while data classified as signal type 4 is compressed with $cr = 80\%$.

These two optimization problems are computed offline, and the relations *{localization, signal type, cr}* are stored in the internal memory of the node in a *Look-up table*.

4.1.5 Measurement Matrix

When the output of the previous module leads to a new $cs[t]$ compression ratio, this is used to select the corresponding *measurement matrix*. Raw data are multiplied by the DCT matrix and then sent wirelessly to the back-end unit.

4.2 Back-End View

4.2.1 Fine-Grained Feature Generation

A segment of compressed data that includes information of the *cr* used is received. With this information, the estimation x' of the original signal is computed using a copy of the measurement matrix A stored on the node. Features are computed from the recovered segment to perform the node localization and the activity recognition.

4.2.2 Fine-Grained Node Localization

In this phase we perform node localization on the received and reconstructed data segments. The *Random Forest* procedure has been chosen as in the sensing node. Once the data are recovered, the system works independently of the compression ratio used in the sensing node. The *Random Forest* is trained using the data recovered for all compression ratios. This leads to a fine-grained node localization. All the 30 features are considered in the *Random Forest* algorithm.

4.2.3 Fine-Grained Activity Recognition

As in Section 4.1.4, after training different algorithms, such as multi-layer perceptrons or random trees, a *Random Forest* algorithm has shown the best performance in terms of training error. After the node localization phase, the fine-grained activity recognition is performed. All features are considered for this purpose in the back-end.

5 Evaluation and Results

In this section, we thoroughly evaluate the proposed system. We present and discuss the results and the performance of

the two optimization problems: i) clustering of the optimum number of types of signals; and ii) optimum assignment of compression ratios for those clusters in addition to the modules in the sensing nodes and the back-end units. Finally, a realistic experimental framework is proposed and used to measure the energy savings of the system.

5.1 Data

To evaluate the framework, the real world human activity dataset *UCI Daily and Sports Activities Data Set* [43] is used. These are real data acquired from 8 subjects (4 male and 4 female, between the ages 20 and 30) asked to perform 19 daily and sports activities, such as: sitting, walking, running at different speed, jumping, rowing or playing basketball. Data are acquired at $25Hz$ using a triaxial accelerometer. Each activity takes 5 minutes and data are chopped into non-overlapping segments of 5 seconds (125 samples per axis). Data were acquired simultaneously from a wearable sensor placed on five different on-body locations: torso (T), right arm (RA), left arm (LA), right leg (RL) and left leg (LL). The data set was split into training (80 percent) and test (20 percent).

From these data, 10 different statistical features were extracted [44] for each axis of the accelerometer. This leads to 30 features per data segment of 5 seconds. These features are: the amplitude of the segment amp, the median med, the mean mn, the maximum and minimum values (max and min), the peak to peak increment $p2p$, the variance and the standard deviation (var and std), the root mean square value rms and, finally, the difference between the first and last values of the sequence $s2e$. In Section 5.5 our feature selection mechanism is described. Furthermore, the impact in energy of this selection is elaborated in Section 5.4.

5.2 Optimization

In this section, the results of the optimization problems using GE and NSGA-II are shown. For both processes, the number of individuals for each generation has been set to 250. Likewise, the number of generations has been set to 1,000 and 500 for signal-type clustering optimization and compression ratio assignment optimization, respectively. For each generation, all individuals are evaluated externally using compiled code created using Matlab. The number of individuals per generation has been chosen to reduce the amount of external calls to the classifiers. The number of generations has been chosen after some tests. For these values of the parameters, the solutions remain approximately stable and no new solutions (individuals) appear at the end of the optimization process.

5.3 Performance

Following paragraphs show the performance results of the optimization problems on-node and the *Random Forest* classifier for both node localization and activity recognition in the sensing node and the back-end unit.

5.3.1 Coarse-Grained Feature Generation

This module computes 10 features for each axis of the accelerometer. Features are calculated over compressed data which allows important reduction of the energy consumption.

5.3.2 Signal-Type Clustering Optimization

Solutions that solve Eq. (8) are plotted in Figs. 7a, 7b, 7c, 7d, and 7e. These figures represent the best solutions (black points)

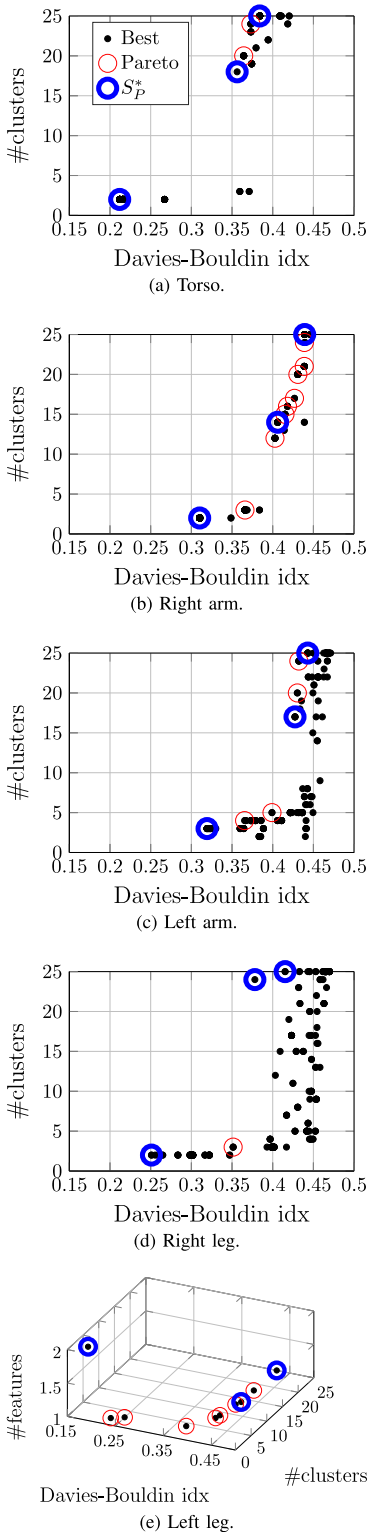


Fig. 7. Signal-type clustering optimization.

of the GA, i.e., the last generation of the population. The non-dominated solutions S_P represent the Pareto front (circles).

Because we have three optimization goals, the figures should be plotted in three dimensions. However, since all the solutions on the Pareto front share the same value on feature axis $\rho_{\#f} = 1$, we plotted the figure in 2D. The only exception is the location *Left leg* that uses two features. All the 10 solutions in Fig. 7e belong to the non-dominated Pareto front.

TABLE 1
Values of Optimized Objectives per Location

Location	f'	$\#k$	$\#k_{merged}$	$\#cr_{unique}$	$\nu_{cluster}(\%)$
T	mnX	18	17	10	58.8
RA	mnX	14	13	8	61.5
LA	$medY$	25	24	13	54.2
RL	mnZ	2	2	2	100
LL	mnZ	14	14	11	78.6

Reducing the number of features is a significant result because it diminishes the complexity of the implementation of the *k-means* algorithm in the sensing node, and the same time maximizes the performance of our resulting system.

It is worth mentioning that the minimization of the number of features was easily accomplished by the NSGA-II algorithm—from the first generations the selection was rapidly established. This can be understood as another indicator that not all the features are useful and only a small selection of them becomes meaningful. In a additional mono-objective optimization process it was observed that the optimal number of clusters for our training set is $\#k_{opt} = 2$. As our criterion to reduce the energy costs goes through increasing this number, the optimization process achieves solutions using a high number of clusters appeared in the Pareto front (in the next section it is shown that the criterion to maximize $\#k$ is correct).

In the Pareto fronts we have selected two extreme solutions and a third one between them to compute the next optimization process. These selected solutions $S^*_{P_{ST}}$ are represented as blue bold circles in Figs. 7a, 7b, 7c, 7d, and 7e. The Davies-Bouldin index $\overline{R}$ remains in the same range for all locations but for *torso*, where it is lower. For all the observations in the heuristic experiments, the values $0.15 \leq \overline{R} \leq 1.60$; so results are considered acceptable being always lower than 0.5, and lower than 0.45 for the selected solutions $S^*_{P_{ST}}$.

Table 1 shows the number of unique compression ratio values $\#cr_{unique}$ for each on-body location. We define the relative clustering-ratio assignment efficiency ($\nu_{cluster}$) for each location as the ratio of the number of unique compression ratio assignments over the number of distinct clusters, as follows:

$$\nu_{cluster} = 100 \frac{\#cr_{unique}}{\#k_{merged}} (\%). \tag{12}$$

Smaller values of $\nu_{cluster}$ indicate more redundancy, meaning that the same compression ratios appear several times while clusters cannot be merged in Fig. 6, because the signal types for that location are quite similar. On the contrary, larger $\nu_{cluster}$ means that a the identified clusters are more varied in terms of the compressed sensing ratios assigned to them, and the look-up table in Fig. 1c is smaller as well. As an example, Table 2 lists the distinct clusters for Torso location and their associated compression ratios. According to the obtained efficiency, we can observe that the granularity criterion established for the number of clusters $\#k$ was computationally sufficient.

5.3.3 Compression Ratio Assignment Optimization

For each solution $S^*_{P_{ST}}$, another NSGA-II optimization has been performed. The Pareto front resulting from each optimization over the training set is shown in Figs. 8a, 8b, 8c, 8d, and 8e.

For the criterion stated in Eq. (11), horizontal lines have been drawn in Figs. 8a, 8b, 8c, 8d, and 8e. These lines cross through the three Pareto fronts and the results are compared in Table 3 for each location. This table compares the accuracy of the baseline approach, our adaptive temporal compressed sensing methodology and three different naive solutions—the minimum, maximum and average value of the selected solution (bold numbers) from all locations. Adjusting ε_{th} changes the optimal compression ratios, respective to the derivative of the Pareto front. In Figs. 8a, 8b, 8c, 8d, and 8e, it can be seen that, a larger value for ε_{th} leads to more compression and energy saving, and lower number of clusters. On the contrary, smaller ε_{th} leads to less compression and energy saving, and a higher number of clusters ($\#k$).

As expected, in most of the cases, a high number of clusters leads to high weighted mean compression ratios, and thus, a lower energy consumption. The compression ratios are different for each node location and similar for few locations—arms and legs—because movements are similar. The highest compression ratios are achieved for nodes located on arms. Therefore, it is expected that nodes placed on arms save more energy due to data transmission.

We note that the same compression ratio could be assigned to different clusters. Those with the same compression ratio that are adjacent are merged in $\#k_{merged}$, and the new centroid is computed. After this process, $\#k$ has been reduced at least in one for three locations (see Table 1).

The highest, the lowest and, the average compression ratios of the solutions selected for each location have been studied in the naive methodology. The last three rows in Table 3 compare their accuracy. Only the solution with $cr = 42.8\%$ is compliant with the quality criterion stated in Eq. (11) (5 percent threshold on residual error of classification). This is the solution selected to compare with our proposed methodology. So, henceforth, the energy consumption due to transmission for the naive case and LL must coincide.

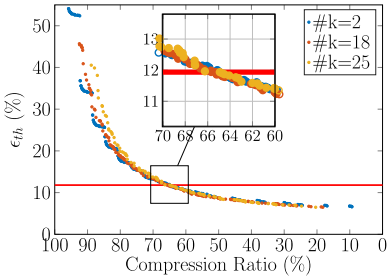
5.3.4 Coarse-Grained Node Localization

For the node localization process performed in the sensing node, a *Random Forest* algorithm was trained. As memory is a limited resource in monitoring devices, only one *Random Forest* model is stored. This model was trained with data without compression from all locations. The location is detected when compressed data are processed by this module. The accuracy of the trained model reaches up to 97.8 percent as shown in the confusion matrix in Table 4.

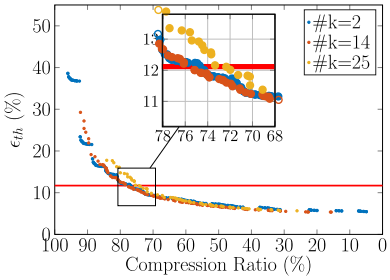
Mislocalizations happen mostly between legs. The higher error happens when 14.4 percent of times LL is classified as

TABLE 2
Clusters for Torso

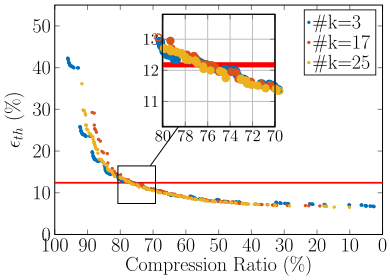
Cluster	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
cr (%)	72	64	72	12	72	44	76	40	68	72	56	64	56	76	88	64	28
Centroid (units)	−6.74	−4.67	−3.72	−3.14	−1.84	−0.59	0.36	1.43	2.51	4.23	6.99	7.69	8.38	8.96	9.25	9.51	9.75



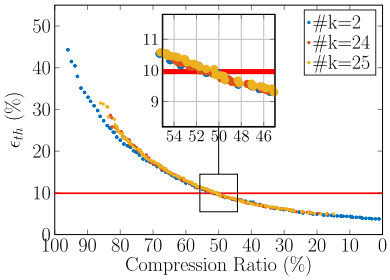
(a) Torso.



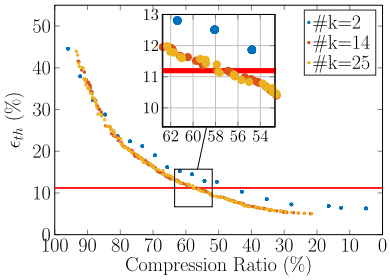
(b) Right arm.



(c) Left arm.



(d) Right leg.



(e) Left leg.

Fig. 8. Compression ratio assignment optimization.

RL and 11.8 percent of times RL is classified as LL. This will lead to over-compression and sub-compression respectively—according to the selected solutions in Table 3. These dissimilarities might compensate each other regarding the accuracy and the energy consumption. Therefore, according to our results in Table 4, we show that this module does not propagate the error because of compensation.

5.3.5 Fine-Grained Node Localization

When data is reconstructed in the back-end, features are computed and node localization is performed. The accuracy of the fine-grained node localization module is shown in the second column in Table 5, and they are the best classification of the node location possible in the back-end if there were no misclassifications in the coarse-grained node localization. These results are compared with the ones achieved using the naive system with a fixed compression ratio of $cr = 42.8\%$ and the baseline approach as well.

A *Random Forest* algorithm has been trained for each approach and the results are shown in Table 5. As can be seen, results are really close each other and there are only slight differences with respect to the baseline. Our approach differs only 1 percent accuracy from the naive methodology—despite the average compression ratio of our approach is 62.6 percent compared with 42.8 percent of the naive method.

5.3.6 Fine-Grained Activity Recognition

The actual results for the activity recognition are those shown in Table 3. $\alpha_{AR_{Baseline}}$ represents the baseline accuracy for the activity recognition algorithms trained in the back-end of the system. As there are no limitations in the computation, in the back-end all features are computed and used. $\alpha_{AR_{Adap.Temp.}}|\overline{cr}_{set,f'}$ represents the final accuracy for the activity recognition for our methodology. The final solutions are those in bold in Table 3. The average $\alpha_{AR_{Adap.Temp.}}|\overline{cr}_{set,f'}$ for all locations s 89.0 percent. As aforementioned, the accuracy of the naive solution selected is 91.0—2 percent higher that the average value of our methodology. In spite of this, in Section 5.4 we show that, being both methodologies compliant with the quality criteria, the benefits of our proposal in terms of energy savings are significant.

5.4 Energy Consumption

To test the energy performance of the proposed methodology, an experimental set-up has been developed. In this section we show the results in terms of energy savings due to the data transmission (which represent the 81 percent of energy consumption in the baseline mode), and we compare them with the small overhead that the system introduces for the computation of the on-node calculations performed in our custom-designed activity monitoring device shown in Fig. 1c.

5.4.1 Experimental Set-Up

In this study, we programmed our methodology onto an actual activity monitoring device (see Fig. 9). The experimental set-up consists of a sensing device that performs the system's tasks and another device to measure its power consumption. The sensing node incorporates one ATmega328 microcontroller running at 8 MHz, a 10-bit precision accelerometer sensor ADXL335, and a Bluetooth module RN41-3 that sends the data to a computer configured in deep sleep mode and using a power transmission of $-12dBm$. The device under measure is a platform that monitors the current in the sensing node's battery using the INA219 sensor. Current is measured through a shunt resistor of 0.1Ω the measurement precision is $100\mu A$ and sent to the computer at a rate of $500Hz$. All results are computed for the weighted average compression ratio achieved for each location in Table 1.

TABLE 3
Accuracy of the Selected Solutions after the Two Optimization Processes and the Naive Approach Compared to the Baseline

Location	T			RA			LA			RL			LL			Average $\alpha(\%)$
$\alpha_{AR_{Base.}}(\%)$ / $\varepsilon_{th}(\%)$	93.2 / 11.8			93.3 / 11.7			92.6 / 12.4			95.1 / 9.9			95.0 / 10.0			93.8
#k	2	**18**	25	2	**14**	25	3	17	**25**	**2**	24	25	2	**14**	25	
$\overline{cr}_{set,f'}(\%)$	64.9	**65.2**	64.6	76.7	**77.4**	75.4	76.0	75.7	**76.6**	**51.2**	50.6	50.3	37.5	**42.8**	40.1	89.0
$\alpha_{AR_{Adap.Temp.}}\ \lvert\overline{cr}_{set,f'}(\%)$	88.2	**88.4**	88.3	88.6	**88.6**	88.2	87.7	87.6	**87.7**	**90.2**	90.2	90.4	90.2	**90.0**	90.0	
$\alpha_{AR_{Naive}}(\%)\ \lvert cr = \mathbf{42.8}\%$	91.9			91.5			92.1			89.9			89.6			**91.0**
$\lvert cr = 62.6\%$	89.0			89.8			90.1			83.6			83.5			87.2
$\lvert cr = 77.4\%$	83.6			86.7			87.7			76.1			76.7			82.2

To obtain the total energy consumption ε_T, a linear energy model is proposed in Eq. (13) and the results are compared with the baseline energy consumption $\varepsilon_{Baseline}$, where the dummy system only transmits raw data every 5 seconds. This model considers the single energy consumption of each module of the system (node side in Fig. 1c) in three main levels: sensing σ, processing π and transmission τ. We assume that the total energy consumption is the sum of individual consumption of each module of the system.

Tasks are executed independently in a loop. The execution time to process 5 seconds of gathered data is also measured using time-stamps. The drawn current is measured as well. Using the timing information we are able to compute the energy consumption

$$\varepsilon_{Total} = \sigma + \pi + \tau, \tag{13}$$

with

$$\pi = SF_\varepsilon + FG_\varepsilon + (NL_\varepsilon + ST_\varepsilon) + MM_\varepsilon. \tag{14}$$

In the naive approach

$$\pi_{Naive} = MM_\varepsilon, \tag{15}$$

because no feature computation, on-body node localization, or signal-type detection is performed.

To ensure that the system passes through all the states, a piece of the test dataset has been coded in program memory. This data is read once and stored in RAM as for real gathered data. To lead to the energy model in Eq. (13), we proceed as follows:

1) To measure the consumption of the sensing process σ, the accelerometer module is connected and a simple code reads movements at $25Hz$—same sampling rate than the UCI Data Set uses—and stores data in RAM.
2) π is the result of the five different blocks in the node side. For each location we individually measure the energy consumption of the processing modules. As shown in Eq. (14): the SF_ε consumption of the *Sparse Filter*, the FG_ε due to the *Coarse-grained Feature Generation*, both machine learning algorithms blocks *Coarse-grained Node Localization* (NL_ε) and *Coarse-grained Signal-Type* detection (ST_ε), and the *Measurements Matrix* using the DCT (MM_ε).
   - SF_ε: For each location is stored in ROM memory. It is a 1D array of $\overline{cr}_{set,f'}$ random numbers. Each number represents an index of one element of 5 seconds of data stored at $25Hz$. Its energy consumption is SF_ε.
   - FG_ε: the consumption of this module depends on the compression ratio, the larger the segment, the heavier the computation. The consumption of this module will be calculated when computing all 30 features.
   - NL_ε: a *Random Forest* is implemented. This is a sequence of $if-else$ sentences. For the implementation using $\#f = 30$, the average size (number of nodes) of the trees is 2020. The number of trees in the forest is 100. For the sake of simplicity we coded one, but the total energy consumption takes it into account.
   - ST_ε: this consumption is the result of a *k-means*algorithm. As the optimal number of features found for all locations is $\#f = 1$, this implementation finds the closer centroid within a 1D group of $\#k_{merged}$ clusters in a sequence of $if-else$ sentences (see Table 1).
   - MM_ε: the multiplications of DCT matrices lead this consumption. The measurement matrices are stored in ROM and loaded to RAM at the beginning of the execution in order to perform the multiplications faster.
3) Finally, the model includes the energy consumption of data transmission using Bluetooth, τ. For these experiments:
   - First, we measure the baseline energy consumption $\varepsilon_{Baseline}$ when our system is not implemented

TABLE 4
Confusion Matrix of the Coarse-Grained Node Localization Module (%)*

Classified as →	T	RA	LA	RL	LL
T	97.8	0.3	1.6	0.0	0.3
RA	2.7	91.8	4.3	1.2	0.0
LA	4.5	1.2	91.2	1.5	1.6
RL	0.0	4.8	0.1	83.3	11.8
LL	0.9	1.5	7.1	14.3	76.1

Light gray means sub-compression. Dark gray means over-compression.

TABLE 5
Accuracy of Node Localization in Back-End Unit (%)

If node placed in	Temporal adaptive compressed sensing	Naive $cr = 42.8\%$	Baseline
Torso	97.4	98.2	98.6
Right arm	97.6	98.7	98.7
Left arm	95.4	97.1	97.5
Right leg	95.1	95.6	97.7
Left leg	93.2	93.8	96.7
Average	95.7	96.7	98.2

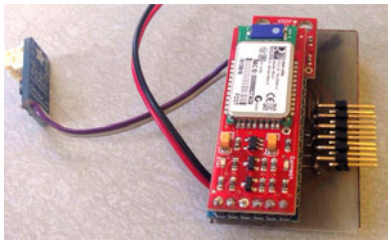


Fig. 9. Experimental monitoring node.

and all raw data is transmitted every 5 seconds. Every second the Bluetooth transmitter wakes up and sends 375 samples of data—25 samples for each axis of the accelerometer using 10 bit precision. Thus, the energy overhead of this baseline solution comes mainly from the switching of the transmitter.

- The amount of samples to send every 5 seconds for each location according to the weighted average compression ratios are: i) 132 for T, ii) 87 for RA, iii) 90 for LA, iv) 183 for RL and v) 216 for LL. Using the naive methodology 216 samples are sent every 5 seconds.

With this experimental set-up, we obtain the energy consumption listed in Table 6. This table shows the average energy values per second. The consumption of the sensing process σ is an offset equal for all locations. π is the average consumption of the processing in the sensing node (see Eqs. (14) and (15)). τ is the consumption due to data transmission that takes the lower values for locations with higher compression ratio.

As expected, the larger the weighted average compression ratio, the higher the energy savings. This can be seen especially when node is located in LA. Fig. 10 shows the relative percentages of the energy consumption for the tree components assumed in Eq. (13), for each location, the naive approach and the baseline. The energy consumption of the most expensive process, the transmission, has been reduced up to 81.2 percent for LA, and a small overhead of our system leads to total energy savings up to 60.6 percent in this location when compared with the baseline. Compared with the naive approach, for the LA location too, these values have been reduced up to 61.5 percent in transmission, and 35.0 percent for the overall system. All these solutions and the naive one are compliant with the restriction of extra error in activity recognition of 5 percent over the baseline.

While the energy of computation might seem expensive, is a small $\sim 13\%$ overhead for all locations. Computation time ranges from 1.9 to 2.2 seconds, from lower to higher

TABLE 6
Average Energy Consumption and Total Energy Savings

Location	$\sigma(mJ)$	$\pi(mJ)$	$\tau(mJ)$	ε_{Total}	$\varepsilon_{Tot.Savings}(\%)$
T	13.4	4.7	15.9	34.0	51.6
RA	13.4	3.5	11.2	28.1	60.0
LA	13.4	3.6	10.7	27.7	60.6
RL	13.4	5.7	23.5	42.6	39.4
LL	13.4	6.8	27.8	48.0	31.7
Naive	13.4	1.4	27.8	42.6	39.4
Baseline	13.4	-	56.9	70.3	-

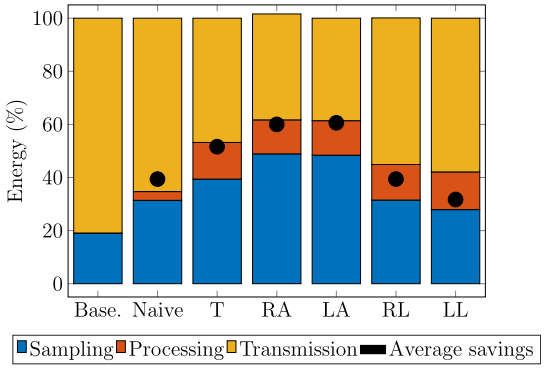


Fig. 10. Relative energy consumption in the sensing node.

$\overline{cr}_{set,f'}$. All consumption that performs the expression of π are detailed in Table 7.

Computation of features is the most expensive process. Computation of the *Sparse Filter* (SF_ε) and *Signal-type clustering optimization* (ST_ε) are almost negligible.

Our analysis estimated computation power overhead of 13 percent (of the total energy consumed, for all locations). This is quite insignificant as at the end, total energy savings of at least, 31.7 percent was achieved. The average saving over all locations in our approach was 48.7 percent which is considerably higher that the naive model (39.4 percent). The only case where naive model outperformed our approach was LL location. The reason behind this observation is that when optimizing the compression ratio for the naive approach, the maximum ratio belonged to LL and hence, for only that on-body location, the naive approach can perform well without the need for an on-node localization module. However, it is obvious that for any other location this saving would degrade due to lack of on-node location-awareness and intrinsic bias of the optimization model towards the LL location. Our result is significant, because we were able to achieve higher savings on average despite having computation overheads (e.g., by having a coarse-grained node localization module) while making the system much more robust against on-body sensor displacement.

We have demonstrated that significant reduction of data transmission of our adaptive temporal compressed sensing approach lead to energy savings, resulting into substantial increments of battery life. We have also demonstrated that the adaptive temporal version is better, on average, than the naive approach compressing data at a fixed rate.

5.5 Study of Features

As mentioned previously, 30 features have been extracted. It was noticed that some features were selected more times than others by the optimization algorithms. An example is shown in the histogram shown in Fig. 11. This figure depicts the distribution of features selected by the Optimization problem #1 for both dominated and not dominated solutions.

A preliminary study has been carried out in order to see correlation between features, because some features might be highly correlated. To see this, Fig. 12 shows the absolute value of the matrix of correlations for all features and each location. As it can be observed, correlation between features is similar among locations. These matrices include all the subjects and activities. The lighter the color, the higher the correlation. We can easily see that there are only few features with low correlation between them; this might help with the simplification of the system.

TABLE 7
Average Processing Energy Consumption per Second π for $\#f = 30$ and $\#f = 6$ Features

Location	$SF_\varepsilon(\mu J)$	$FG_\varepsilon(mJ)$		$NL_\varepsilon(mJ)$		$ST_\varepsilon(\mu J)$	$MM_\varepsilon(mJ)$	$\pi_{\#f=30}(mJ)$	$\pi_{\#f=6}(mJ)$	$\varepsilon_{\pi_{\frac{6}{30}Savings}}(\%)$
		f_{30}	f_6	f_{30}	f_6					
T	4.0	3.4	2.9	0.5	0.6	0.2	0.8	4.7	4.3	8.5
RA	2.8	2.4	2.0	0.5	0.6	0.2	0.6	3.5	3.2	8.6
LA	2.9	2.5	2.2	0.5	0.6	0.2	0.6	3.6	3.4	5.6
RL	5.9	4.1	3.2	0.5	0.6	0.1	1.1	5.7	4.9	14.0
LL	6.9	4.9	3.8	0.5	0.6	0.1	1.4	6.8	5.8	14.7
Naive	6.9	-	-	-	-	-	1.4	-	-	-

According to results in Fig. 12, and results in Table 1, we observe that i) only 1 feature is necessary for the signal-type clustering, and ii) only few features are not highly correlated.

Low correlated features in Fig. 12 are mostly those chosen by the optimization process—median med and mean mn are well represented. It is interesting to see that the $s2e$ values, although they are not correlated with any other feature, are barely selected. We can explain this fact as the information provided by those signals is not enough to distinguish the signal type.

As can be seen, the selected features in Table 1 match with the expected ones regarding the distribution in Fig. 11 too. So, we could reduce the number of features to make more efficient the coarse-grained node localization and the signal-type selection on sensing nodes.

For the sake of simplicity, this paper only presents a brief study to show the feasibility of feature reduction. To do this we have reduced the number of features down to 6 (median and mean for all axes), and the study of the impact on energy savings due to data processing is shown in Table 7. In this table, it can be seen that using $\#f = 6$ consumption values of FG_ε and NL_ε vary. Node localization using less features leads to a higher consumption of NL_ε as expected because using less variables makes the classification harder and the trees are larger (2,467 nodes on average).

In this example, we showed that by further reducing the overhead of computation, up to 14.7 percent ($1mJ$) extra energy savings were reached. The overhead due to computation—although still relatively small—can reach 14.2 percent of total energy consumption (see Fig. 10), and decrementing this value in $1mJ$ further helps to improve the battery life.

A more intelligent feature selection will draw better results, reducing the maximum gap allowed of extra error. However feature selection techniques require a deep knowledge of data. According to the results drawn in previous figures, it seems that it is worthwhile to calculate the trade-off between accuracy loss and energy savings. We aim to tackle this issue in our future work from an energy efficiency perspective.

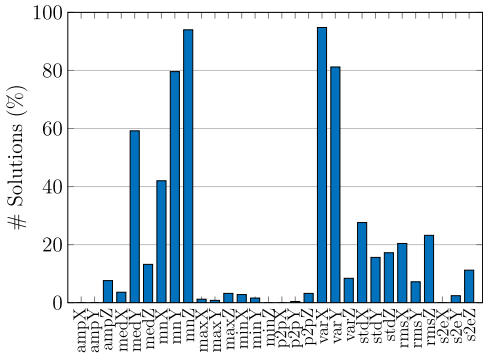


Fig. 11. Histogram of features selected for the best GE solutions in the optimization problem #1.

5.6 Comparison with On-Node Activity Recognition Alternatives

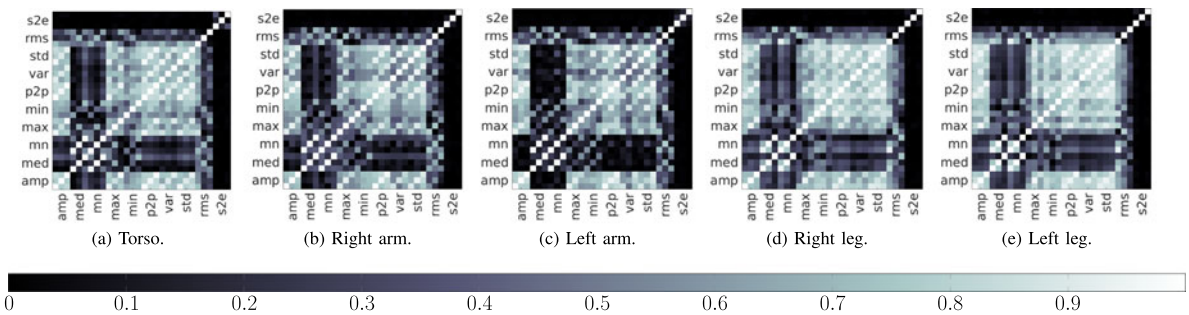
On the contrary to the problem discussed all along this work, there might be applications where the motion data are not necessary to be transmitted and activity recognition is preferred to be done on sensing node and only the recognized activity is sent to the back-end. The assumptions in these applications are totally different from ours. These will benefit of a low power consumption mainly due to the reduction of data transmission, but motion data will not be available in the back-end for other purposes—such as visualization on further computation.

On-node activity recognition can be implemented in different scenarios: i) over uncompressed data or ii) over compressed data. There will be three major processing modules in the architecture of scenario (i): fine-grained feature generation, fine grained node localization and fine grained-activity recognition. The architecture of scenario (ii) would be similar to the one proposed for our adaptive temporal compressed sensing methodology in Fig. 1c, but substituting the *Measurement Matrix* for a coarse-grained activity recognition module. Both scenarios remove all the processing modules in the *Extreme-end unit*. In the following lines we show the energy consumption and savings of these two alternatives. The accuracy of these implementations will be commented as well as shown in Table 8.

On-body node localization and on-node activity recognition can be implemented using *Random Forest* algorithms on the sensing node. Please, note the reader that 30 features are extracted/generated for further processing on-node, and so, henceforth we assume that size and consumption of the trained Random Forest algorithms for both processes—on-body node localization and on-node activity recognition—-are similar and both consume $0.5\ mJ$ as seen in Table 7 (an extra consumption overhead of $0.5mJ$ is imposed by the on-node activity recognition).

Consumption due to data transmission is $0.1\ mJ$ in both scenarios. This is negligible compared with the consumption of our approach due to only the label of the activity is sent to the back-end once every 5 seconds (to be comparable with our results).

Scenario (i): if feature extraction is carried out over uncompressed data, the feature extraction process will consume considerably more energy. In this scenario the consumption is the same independently of the localization. The average total energy saving increases 17.2 points from 48.7 to 65.9 percent compared to our proposal. The average accuracy, will be the one expected for the Baseline solution 98.2 percent (see Table 5).



(a) Torso. (b) Right arm. (c) Left arm. (d) Right leg. (e) Left leg.

Fig. 12. Absolute value of the matrix of correlation of features for each location considered. The lighter the color, the higher the correlation.

Scenario (ii): the feature extraction module in this scenario consumes the same than in ours. The average energy saving is 74.4 percent, which is considerably higher than the ones achieved for our proposal and on-node dine-grained AR. However, when we apply activity recognition over sparse data (i.e., coarse-grained AR), we should expect a reduction in the accuracy. Table 8 lists a comparison of accuracy of AR (α_{AR}). Accuracy drops down to 88.2 percent, and it is worthwhile to mention that this level does violate our 5 percent threshold on accuracy decline (remember the reader that our adaptive temporal compressed sensing methodology reaches 95.7 percent average accuracy in the back-end).

As aforementioned, the comparison of the methodology defended in this paper with the two alternatives that implement on-node activity recognition is not fair, as the assumptions and applications are totally different. The achieved increments in energy saving are accompanied by loosing the original motion data in the back-end, and a possible reduction in the accuracy.

6 Discussion

A variation of the proposed system can be developed without on-node computation where the signal type detection and on-body node localization are performed on the back-end and the feedback are transmitted to the sensing node for adjustment of compression ratio. The main benefit of this alternative approach is the elimination of processing overhead on the local node. However, the additional cost of constant transmission of feedback from back-end to the sensing node must be taken into account. In addition, the practical limitations that this alternative approach poses on the system makes it less favorable in many real-world applications of wearable activity recognition. The constant and real-time dependability of this approach is often not affordable. There are two reasons for this argument:

i) Assuming constant connectivity is too optimistic. Many wearable devices rely on close-range, low-cost communication technologies such as BLE, and are operated by humans in highly mobile settings. These system often buffer the data, when out of range of the central node, and transmit it once the central node become reachable.

ii) In practical settings, frequent sensor data/feedback transmission is not favorable due to its large power consumption. Many of more recent wearable technologies such smart-watches offer multiple means of connectivity such as LTE, WiFi and BLE. Efficient applications should aim to maximize their transmission on less costly links by locally storing data until a reliable and cheap connection become available. As a result, we argue that, with comparable overall energy cost, the proposed approach is far more practical than the aforementioned alternative because it does not need to transmit sensor data (since it does not rely on external feedback).

TABLE 8
Energy Consumption, Energy Savings and Accuracy for On-Node AR Scenarios

On-node AR	Location	$\sigma(mJ)$	$\pi(mJ)$	$\tau(mJ)$	ε_{Total} (mJ)	$\varepsilon_{Tot.Sav.}$ (%)	$\alpha_{AR}(\%)$
Coarse-grained	T	13.4	4.4	0.1	17.9	74.5	91.0
	RA	13.4	3.4	0.1	16.9	76.0	84.1
	LA	13.4	3.5	0.1	17.0	75.8	91.2
	RL	13.4	5.1	0.1	18.6	73.5	87.0
	LL	13.4	5.9	0.1	19.4	72.4	87.6
Fine-grained	All	13.4	10.5	0.1	24.0	65.9	98.2

In our methodology we stated a 5 percent extra error as a quality criterion proportional to the original baseline accuracy for activity recognition (AR). The parameter ϵ_{th} in Eq. (11) is defined to ensure an acceptable performance and avoid over-minimization of sensing rates at the cost of end-result accuracy. It can be viewed as a tuning parameter that can be adjusted by an inference drawn from the domain knowledge (e.g., given the accuracy of the baseline activity recognition algorithm, what is the lowest acceptable accuracy that your application will consider acceptable?). It is worthwhile to mention that larger thresholds will result in more power optimization but will allow for significant performance decline. On the other hand, excessively small thresholds will not allow for significant data sensing optimization and therefore are not desirable in energy stringent applications.

7 Conclusion

The proposed methodology solves, through a novel adaptive temporal compressed sensing technique, an important problem of the monitoring devices in the paradigms of the MCC and RHM: battery consumption due to excessive wireless transmissions. In this work we apply our methodology in a physical activity recognition problem, but the system is extensible to any other application of the IoT where power efficiency is an obstacle. Utilizing metaheuristic optimization techniques based on GE, our proposal achieves energy savings in transmission of up to 81.2 percent, with negligible energy overhead in the monitoring devices, which leads to global savings of up to 61.0 percent.

The proposed framework performs coarse-grained on-body sensor localization and unsupervised clustering algorithms are employed to autonomously reconfigure compressed sensing ratios ranging from 42.8 to 77.4 percent on average for different on-body node localization. With this approach, we achieve significant accuracy levels between 87.7 and 90.2 percent when performing fine-grained activity recognition in the back-end computing unit (e.g., a smartphone or a data center).

The proposed optimized adaptive temporal compressed sensing methodology has a bounded error of 5 percent over the baseline, and reaches higher energy savings when compared with the state-of-the-art: a naive compressed sensing approach with invariable compression ratio. A coarse-grained on-body sensor localization—based on a *Random Forest*—is performed and it has been shown that the error in node localization is not propagated to the fine-grained activity recognition in the back-end unit.

An additional study about the reduction of the number of features has been carried out. Preliminary results showed that, after our feature optimization, we can achieve up to 14.7 percent energy savings in computation, which further contributes to an extended battery life.

Acknowledgments

This work was supported in part by the EU (FEDER) and the Spanish Ministry of Economy and Competitiveness under Research Grants TIN2015-65277-R and TEC2012-33892 and the US National Science Foundation under grant CNS-1566359. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding organizations.

References

[1] Y.-L. Zheng, X.-R. Ding, et al., "Unobtrusive sensing and wearable devices for health informatics," *IEEE Trans. Biomed. Eng.*, vol. 61, no. 5, pp. 1538–1554, May 2014.

[2] P. Kugler, D. Schuldhaus, U. Jensen, and B. Eskofier, "Mobile recording system for sport applications," in *Proc. Int. Symp. Comput. Sci. Sport*, 2011, pp. 67–70.

[3] H. Alemdar and C. Ersoy, "Wireless sensor networks for healthcare: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2688–2710, 2010.

[4] B. G. Steele, L. Holt, B. Belza, S. Ferris, S. Lakshminaryan, and D. M. Buchner, "Quantitating physical activity in COPD using a triaxial accelerometer," *Chest J.*, vol. 117, no. 5, pp. 1359–1367, 2000.

[5] J. Klucken, J. Barth, et al., "Unbiased and mobile gait analysis detects motor impairment in Parkinson's disease," *PloS One*, vol. 8, no. 2, 2013, Art. no. e56956.

[6] P. Rashidi and A. Mihailidis, "A survey on ambient-assisted living tools for older adults," *IEEE J. Biomed. Health Informat.*, vol. 17, no. 3, pp. 579–590, May 2013.

[7] R. Saeedi and H. Ghasemzadeh, "AutoLocate: A machine learning approach for automatic localization of wearable sensors in smart health applications," Academic Showcase at Washington State University, Pullman, WA, 2014.

[8] R. Saeedi, J. Purath, K. Venkatasubramanian, and H. Ghasemzadeh, "Toward seamless wearable sensing: Automatic on-body sensor localization for physical activity monitoring," in *Proc. 36th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2014, pp. 5385–5388.

[9] S. Bhattacharya and N. D. Lane, "From smart to deep: Robust activity recognition on smartwatches using deep learning," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops*, 2016, pp. 1–6.

[10] U. Maurer, A. Rowe, A. Smailagic, and D. Siewiorek, "Location and activity recognition using eWatch: A wearable sensor platform," in *Ambient Intelligence in Everyday Life*. Berlin, Germany: Springer, 2006, pp. 86–102.

[11] D. J. Patterson, D. Fox, H. Kautz, and M. Philipose, "Fine-grained activity recognition by aggregating abstract object usage," in *Proc. IEEE Int. Symp. Wearable Comput.*, 2005, pp. 44–51.

[12] T. Choudhury, G. Borriello, et al., "The mobile sensing platform: An embedded activity recognition system," *IEEE Pervasive Comput.*, vol. 7, no. 2, pp. 32–41, Apr.–Jun. 2008.

[13] D. Akimura, Y. Kawahara, and T. Asami, "Compressed sensing method for human activity sensing using mobile phone accelerometers," in *Proc. 9th Int. Conf. Netw. Sens.*, 2012, pp. 1–4.

[14] A. N. Khan, M. Kiah, et al., "Towards secure mobile cloud computing: A survey," *Future Generation Comput. Syst.*, vol. 29, no. 5, pp. 1278–1299, 2013.

[15] M. Santambrogio, et al., "Power-awareness and smart-resource management in embedded computing systems," in *Proc. Int. Conf. Hardware/Softw. Codes. Syst. Synthesis*, 2015, pp. 94–103.

[16] S. Khalifa, G. Lan, M. Hassan, A. Seneviratne, and S. K. Das, "HARKE: Human activity recognition from kinetic energy harvesting data in wearable devices," *IEEE Trans. Mobile Comput.*, vol. 17, no. 6, pp. 1353–1368, Jun. 2018.

[17] G. Lan, D. Ma, W. Xu, M. Hassan, and W. Hu, "CapSense: Capacitor-based activity sensing for kinetic energy harvesting powered wearable devices," in *Proc. 14th EAI Int. Conf. Mobile Ubiquitous Syst. (MobiQuitous'17)*, 2017, pp. 110–119.

[18] H. Mamaghanian, et al., "Compressed sensing for real-time energy-efficient ECG compression on wireless body sensor nodes," *IEEE Trans. Biomed. Eng.*, vol. 58, no. 9, pp. 2456–2466, Sep. 2011.

[19] S.-W. Chen and S.-C. Chao, "Compressed sensing technology-based spectral estimation of heart rate variability using the integral pulse frequency modulation model," *IEEE J. Biomed. Health Informat.*, vol. 18, no. 3, pp. 1081–1090, May 2014.

[20] Y. Liu, M. De Vos, and S. Van Huffel , "Compressed sensing of multichannel EEG signals: The simultaneous cosparsity and low-rank optimization," *IEEE Trans. Biomed. Eng.*, vol. 62, no. 8, pp. 2055–2061, Aug. 2015.

[21] P. K. Baheti and H. Garudadri, "An ultra low power pulse Oximeter sensor based on compressed sensing," in *Proc. 6th Int. Workshop Wearable Implantable Body Sens. Netw.*, 2009, pp. 144–148.

[22] R. Fallahzadeh, J. P. Ortiz, and H. Ghasemzadeh, "Adaptive compressed sensing at the fingertip of internet-of-things sensors: An ultra-low power activity recognition," in *Proc. Des. Autom. Test Eur. Conf. Exhib.*, 2017, pp. 996–1001.

[23] M. A. Razzaque, et al., "Energy-efficient sensing in wireless sensor networks using compressed sensing," *Sens.*, vol. 14, no. 2, pp. 2822–2859, 2014.

[24] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, Mar. 2008.

[25] Y. C. Eldar and G. Kutyniok, *Compressed Sensing: Theory and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2012.

[26] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[27] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[28] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM J. Comput.*, vol. 24, no. 2, pp. 227–234, 1995.

[29] X. Yuan, J. Yang, P. Llull, X. Liao, G. Sapiro, D. J. Brady, and L. Carin, "Adaptive temporal compressive sensing for video," in *Proc. 20th IEEE Int. Conf. Image Process.*, 2013, pp. 14–18.

[30] S.-Y. Chiu, H. H. Nguyen, R. Tan, D. K. Yau, and D. Jung, "JICE: Joint data compression and encryption for wireless energy auditing networks," in *Proc. 12th Annu. IEEE Int. Conf. Sens. Commun. Netw.*, 2015, pp. 453–461.

[31] M. F. Duarte, M. A. Davenport, D. Takbar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk, "Single-pixel imaging via compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 83–91, Mar. 2008.

[32] J. Constantin, A. Dogan, O. Andersson, P. Meinerzhagen, J. Rodrigues, D. Atienza, and A. Burg, "An ultra-low-power application-specific processor with sub-VT memories for compressed sensing," in *Proc. IFIP/IEEE Int. Conf. Very Large Scale Integr.-Syst. Chip*, 2012, pp. 88–106.

[33] S. Yang and M. Gerla, "Energy-efficient accelerometer data transfer for human body movement studies," in *Proc. IEEE Int. Conf. Sens. Netw. Ubiquitous Trustworthy Comput.*, 2010, pp. 304–311.

[34] C. Ryan and M. O'Neill, "Grammatical evolution: A steady state approach," *Late Breaking Papers Genetic Program.*, vol. 1998, pp. 180–185, 1998.
[35] J. Pagán, J. L. Risco-Mart ín, J. M. Moya, and J. L. Ayala, "Grammatical evolutionary techniques for prompt migraine prediction," in *Proc. Genetic Evol. Comput. Conf.*, 2016, pp. 973–980.
[36] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
[37] J. M. Colmenar, J. L. Risco-Martin, D. Atienza, and J. I. Hidalgo, "Multi-objective optimization of dynamic memory managers using grammatical evolution," in *Proc. 13th Annu. Conf. Genetic Evol. Comput.*, 2011, pp. 1819–1826.
[38] D. L. Davies, et al., "A cluster separation measure," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, no. 2, pp. 224–227, Apr. 1979.
[39] HERO, "Heuristic optimization library," 2016. [Online]. Available: https://github.com/jlrisco/hero
[40] E. Candes and J. Romberg, "l1-magic: Recovery of sparse signals via convex programming," vol. 4, 2005, Art. no. 14. [Online]. Available: www.acm.caltech.edu/l1magic/downloads/l1magic.pdf
[41] M. Hall, E. Frank, et al., "The WEKA data mining software: An update," *ACM SIGKDD Explorations Newslett.*, vol. 11, no. 1, pp. 10–18, 2009.
[42] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
[43] B. Barshan, et al., "Recognizing daily and sports activities in two open source machine learning environments using body-worn sensor units," *Comput. J.*, vol. 57, no. 11, pp. 1649–1667, 2014.
[44] H. Ghasemzadeh, N. Amini, et al., "Power-aware computing in wearable sensor networks: An optimal feature selection," *IEEE Trans. Mobile Comput.*, vol. 14, no. 4, pp. 800–812, Apr. 2015.

Josué Pagán received the MSc (Honors) degree from the Technical University of Madrid, in 2013. He is working toward the PhD degree at the Complutense University of Madrid and works as a researcher at the Technical University of Madrid. His work focuses on the development of robust methodologies for information acquisition in biophysical and critical scenarios. He has worked developing models for prompt prediction and classification of neurological diseases. In 2016, he worked as a researcher with the Embedded Pervasive Systems Lab, Washington State University, under the supervision of Prof. Hassan Ghasemzadeh. In 2015, he worked as a researcher with the Pattern Recognition Lab., Friedrich Alexander University. He is a student member of the IEEE.

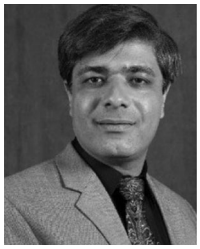
Ramin Fallahzadeh received the BS degree in computer engineering from the Sharif University of Technology, Tehran, Iran, in 2014. Currently, he is working toward the PhD degree in computer science at Washington State University. His current research interests include smart-health, pervasive computing, machine learning, and wireless sensor networks. The focus of his research is on algorithm design and power optimization of networked wearable sensors with applications in healthcare. He is a student member of the IEEE.

Mahdi Pedram received the BS degree in computer engineering from the Amirkabir University of Technology, Tehran, Iran, in 2014. Currently, he is working toward the PhD degree with Advanced Graduate Standing status in computer engineering at Washington State University. He has been working as a research assistant in the Embedded & Pervasive Systems Lab since January, 2016. His research interests include body sensor networks, pervasive computing in healthcare, and embedded system design. He is a student member of the IEEE.

José L. Risco-Martín is an associate professor with the Computer Architecture and Automation Department, Complutense University of Madrid, Spain. His research interests focus on design methodologies for integrated systems and high-performance embedded systems, including new modeling frameworks to explore thermal management techniques for Multi-Processor System-on-Chip, novel architectures for logic and memories in forthcoming nano-scale electronics, dynamic memory management and memory hierarchy optimization for embedded systems, Networks-on-Chip interconnection design, and low-power design of embedded systems.

José M. Moya received the MSc and PhD degrees in telecommunication engineering from the Technical University of Madrid, Spain, in 1999 and 2003, respectively. He is currently an associate professor with the Department of Electronic Engineering, Technical University of Madrid. He is member of the HiPEAC European Network of Excellence. He has served as a TPC member of many conferences, including DATE, IEEE ICC, IEEE MASS, etc. He has participated in a large number of national research projects and bilateral projects with industry, in the fields of embedded system design and optimization, and security optimization of embedded systems and distributed embedded systems. His current research interests focus on proactive and reactive thermal-aware optimization of data centers, and design techniques and tools for energy-efficient compute-intensive embedded applications.

José L. Ayala received the MSc and PhD degrees in telecommunication engineering from the Technical University of Madrid, in 2001 and 2005, respectively. He is currently an associate professor with the Computer Architecture and Automation Department, Complutense University of Madrid. He is a member of the HiPEAC European Network of Excellence, IEEE, ACM, IFIP 10.5, and the CEDA. He has organized several international events as general and program chair, such as VLSI-SoC. He has served as a TPC member of many conferences, including DATE, DAC, ICCAD, etc. His current research interests focus on thermal and energy aware design, design of embedded processors, thermal estimation, 3D integration, health monitoring, and wireless sensor networks. He is a senior member of the IEEE.



Hassan Ghasemzadeh received the BSc degree from the Sharif University of Technology, Tehran, Iran, in 1998, the MSc degree from the University of Tehran, Tehran, Iran, in 2001, and the PhD degree from the University of Texas at Dallas, Richardson, Texas, in 2010, all in computer engineering. He was on the faculty of Azad University from 2003-2006 where he served as founding chair of the Computer Science and Engineering Department at the Damavand branch, Tehran, Iran. He spent the academic year 2010-2011 as a postdoctoral fellow with the West Wireless Health Institute, La Jolla, California. He was a research manager with the UCLA Wireless Health Institute in 2011-2013. Currently, he is an assistant professor with the School of Electrical Engineering and Computer Science, Washington State University, Pullman, Washington. The focus of his research is on algorithm design and system level optimization of embedded and pervasive systems with applications in healthcare and wellness. He is a senior member of the IEEE.