Adaptive Compressed Sensing at the Fingertip of Internet-of-Things Sensors: An Ultra-Low Power Activity Recognition

Ramin Fallahzadeh*, Josue Pagan Ortiz† and Hassan Ghasemzadeh*
*School of Electrical Engineering and Computer Science
Washington State University, Pullman, Washington 99164-2752
Email: {rfallahz,hassan}@eecs.wsu.edu
†Department of Computer Architecture and Automation
Complutense University of Madrid, Madrid, Spain
Email: jpagan@ucm.es

***Abstract*—With the proliferation of wearable devices in the Internet-of-Things applications, designing highly power-efficient solutions for continuous operation of these technologies in life-critical settings emerges. We propose a novel ultra-low power framework for adaptive compressed sensing in activity recognition. The proposed design uses a coarse-grained activity recognition module to adaptively tune the compressed sensing module for minimized sensing/transmission costs. We pose an optimization problem to minimize activity-specific sensing rates and introduce a polynomial time approximation algorithm using a novel heuristic dynamic optimization tree. Our evaluations on real-world data shows that the proposed autonomous framework is capable of generating feedback with $+80\%$ confidence and improves power reduction performance of the state-of-the-art approach by a factor of two.**

I. Introduction

Internet of Things (IoT) is defined as a large set of connected application-specific embedded devices that operate in conjunction with and in service of people. Wearables as a rapidly growing component of IoT have gained much attention recently due to their incredible potentials to enable emerging applications of high societal and economic impacts. Wearables have been wildly utilized for monitoring individuals' physical conditions (physiological attributes, physical activities, etc.) to improve health and wellbeing of their users [1], [2], [3]. Despite the promising future, the utility of these systems is currently limited in part because of the challenges associated with the form-factor and battery lifetime leading to poor integration and user adherence [4]. Similar to prolonged battery life-time, smaller form-factor indicates small batteries thus warranting novel power optimization strategies. While wearables share the same technology as state-of-the-art smartphones, their stringent energy resources requires developing system-level solutions and computationally simple embedded software to overcome the barriers.

Compressed sensing [5], as one of the most recent breakthroughs in signal processing, allows physical entities, under certain assumptions, to be recorded at sub-Nyquist rates without losing considerable information. The theory has been rapidly used and adopted by the researchers in various applications such as multi-media processing and biophysical sensing. Researches in bio-sensing has also studied the application of compressed sensing in recording and compressing electroencephalogram (EEG), electrooculogram (EOG), and electrocardiogram (ECG) signals [6].

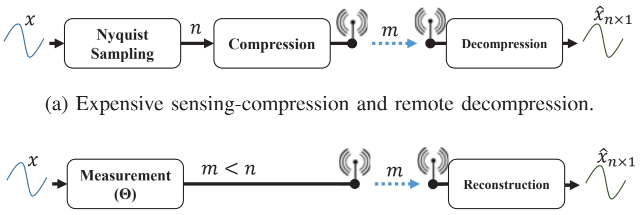
The utility of compressed sensing in IoT and activity recognition, in particular, is substantially limited. Akimura et. al. [7] showed applicability of compressed sensing in sub-Nyquist sampling of physical activities using smartphones. In [8], wearables were used for activity recognition where several types of physical activities were individually compressed and reconstructed remotely, with a normalized root mean square error below 0.05, to achieve higher total power savings. Nonetheless, these approaches did not take into account the inherent need for on-node adaptive compressed sensing, a crucial aspect of adaptive compressed sensing for real-time activity monitoring.

In this paper, we present an ultra-low power wearable compressed-sensing-based activity recognition framework that aims to find an optimal trade-off between compression ratio of each activity type and the overall performance of the activity recognition system and compare the achieved saving to the state-of-the-art compressed sensing approaches. Moreover, we use a novel coarse-grained activity recognition module to generate feedback for the framework autonomously and without need for back-end signal reconstruction process which further eliminates the burden of real-time communication to a back-end server.

II. Preliminaries & Motivation

A. Compressed Sensing

According to the *Nyquist sampling theorem*, for a discrete-time signal to capture all the information from a bound-limited continuous-time signal, the sampling rate must be at least twice as much as the highest frequency of the original signal. However, the introduction of the compressed sensing theory, changed this sampling paradigm. The theory states that under a certain sparsity and incoherence condition, it is possible to efficiently reconstruct a continuous signal with significantly less samples. If signal x can be sampled at Nyquist rate n, compressed sensing is able to use m measurements to represent the signal with minimal information loss where $m < n$. The conversion is performed according to $y = \phi x$ using a sensing matrix ($\phi \in R^{m \times n}$). Furthermore, for the signal x to be applicable for use in compressed sensing, we must find a transformation of x in which the transformed



(a) Expensive sensing-compression and remote decompression.

(b) Low cost compressed sensing and remote reconstruction.

Fig. 1: Traditional sensing-compression vs compressed sensing.

vector is a sparse representation of x. Let $u \in R^n$ be such a representation and $\psi \in R^{n\times n}$ the transformation basis ($x = \psi u$). Therefore, we have:

$$y = \phi\psi u = \Theta u \tag{1}$$

where Θ is referred to as the measurement matrix. As it can be observed, the entire compressed sensing can be computed on the sensor node by simple matrix operations. The incoherence condition states that ψ and ϕ must be incoherent and can be satisfied by choosing a Gaussian random matrix as our sensing matrix. As shown in (1), this is an under-determined formulation (i.e., $m < n$) and therefore, no unique answer can be found for u for a given y. However, under the sparsity assumption, we can find the most sparse answer. Yet, a sparsity maximization problem (formulated as l_0 norm minimization) is hard to solve [9]. The problem can be relaxed to l_1 norm minimization as follows:

$$u = min\ ||u||_1 \quad s.t. \quad ||y - \Theta u|| \ < \ \epsilon \tag{2}$$

where ϵ is a margin for reconstruction error. The solution u can be found in polynomial time using linear programming. Therefore, the original signal will be reconstructed using $\hat{x} = \psi u$. We note the reconstruction process is highly expensive; therefore it needs to be carried out in the receiver-end with much more resource and computation power. Fig. 1 shows an overview of compressed sensing concept compared to the traditional Nyquist sensing.

B. Activity Recognition

The acceleration data collected from physical activities are sparse enough when represented by discrete cosine transform [7]. This transformation has been previously used for sparse representation of audio signals [6]. As a result, compressed sensing can be applied to reduce the sampling rate and consecutively decrease the transmission overhead resulting in significant power saving. In Fig. 2, the blue signal shows $3s$ of walking data collected from one axis of an accelerometer at $25Hz$. Multiplying this signal by discrete cosine basis matrix gives us the discrete cosine transformation which is sufficiently sparse as shown in red in Fig. 2. Randomized sampling of only 66% of the original (blue) signal was reconstructed in the receiver-end (plotted in green). The reconstructed signal ($\hat{x}$), as evident in the figure, is very close to the original signal with normalized root mean square error (nrmse) of 0.08.

The dynamic nature of physical activities and their sparsity has been shown previously [8]. Therefore, varied compression ratio per activity type can be achieved with similar information loss measured by nrmse. Activity-specific optimization of the compression ratio in compressed sensing based activity recognition can potentially result in signification energy saving. Typically, any nrmse below 0.1 is acceptable; however, this cannot necessarily be generalized to our application of interest (i.e., physical activity classification). If two activities are very similar, even small nrmse can lead to significant loss in performance of our classifier. Moreover, for some less similar movements, higher compression can be gained without losing significant classification accuracy. Fig. 3 shows an example where two movements are randomly sampled and reconstructed at 75% compression ratio. While the nrmse may not be generally acceptable (0.16 in Fig. 3a and 0.25 in Fig. 3b), the two activities are still (even visibly) distinguishable. An optimal trade-off between sensing compression ratio of individual movements and the overall performance of the activity recognition system can greatly reduce the sensing and transmission cost in wearable activity monitoring sensors.

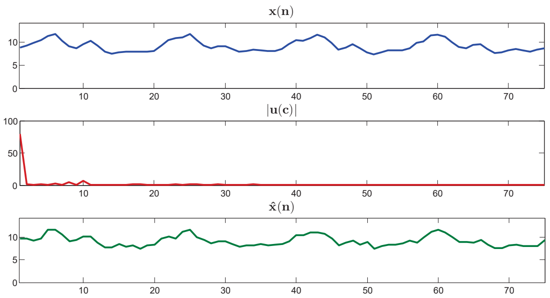
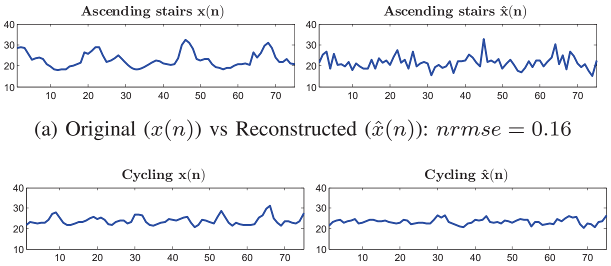


Fig. 2: Sparse recovery of activity 'Walking' with compression ratio of 33% and $nrmse = 0.08$. The blue is the original signal, red is the sparse DCT representation, and green is the reconstructed signal.



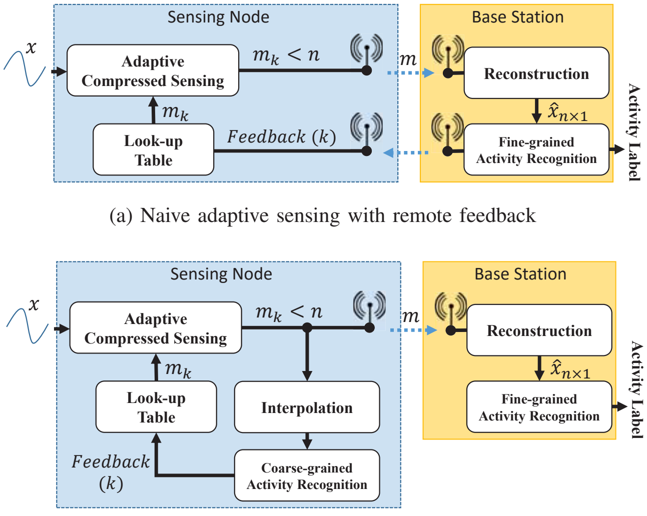
(a) Original ($x(n)$) vs Reconstructed ($\hat{x}(n)$): $nrmse = 0.16$

(b) Original ($x(n)$) vs Reconstructed ($\hat{x}(n)$): $nrmse = 0.25$

Fig. 3: Sparse recovery of two physical activities with compression ratio of 75% causing the binary classifier to lose only 2% accuracy.

III. Minimum Activity-Specific Sensing

We present a novel adaptive compressed sensing framework for activity recognition. As discussed, the maximum compression ratio with respect to the desired activity classification performance varies among different activity types. Assume that the Nyquist rate is n for an activity recognition system comprised of $|A|$ physical activities. Furthermore, let



(a) Naive adaptive sensing with remote feedback

(b) The proposed on-node adaptive compressed sensing

Fig. 4: Overview of the adaptive CS framework

.

E be the overall residual classification error of the activity recognition system sensing at rates $m \in M$. We define the set $M = \{m_1, \ldots, m_{|A|}\}$ containing the compressed sensing rates (m) of each $a_k \in A$. An optimal sensing can be achieved by finding the trade-off between M and E. We formulate this problem and propose a solution in Section III-III-A. Given the solution (i.e., set M), we store the solution locally in a look-up table. The query to this look-up table is the activity ID (k) and the output is the corresponding measurement size (m_k) that is fed to the adaptive compressed sensing module, as shown in Fig. 4. The query is made through the feedback from an activity recognition module that detects the changes in activity type and initiates the process for adaptive change in the value of m.

Conventionally, such adaptive framework requires real-time feedback from base station where the feedback is generated as utilized by researches in adaptive biosignal sensing [6]. This requires the sensing node to be continuously communicating with the base station. Apart from its communication overhead, this approach is infeasible because it requires the sensing node to be always connected to the cloud. The reason for this choice of design is that the feedback generation requires that we first reconstruct the signal. Since reconstruction is a resource consuming process, it cannot be carried out on the node with limited power and computation resources. Alternatively, we propose a low-power on-node feedback generator that uses a coarse-grained activity recognition module to provide real-time feedback to our system. As a result, our framework can perform autonomously on the node and transmit the optimally sensed acceleration data to the base station in an opportunistic fashion. This not only eliminates the requirement for real-time connectivity but also effectively reduces the communication overhead. Fig. 4 illustrates and overview of the state-of-the-art approach versus the proposed framework.

A. *Problem Formulation*

We aim to minimize the number of measurements ($m_k \in M$) for each class of activity ($a_k \in A$) such that the decline in performance of activity recognition is bounded by a given threshold (β). This problem can be formulated as follows.

$$Minimize \quad ||M||_{l1} \tag{3}$$

Subject to

$$E_O(n, M) \leq \beta \tag{4}$$

where $E_O(.,.)$ is the overall error introduced to our multi-label activity classification system using sensing rates $m \in M$ compared to the theoretical optimum of Nyquist rate n. This problem has several properties: (1) it is a nonlinear discreet optimization problem; this is clear because the constraint in (4) is nonlinear; (2) the problem is composed of a non-separable set of constrained optimization problems. It translates to minimizing each m_k with some unknown threshold $\beta_k \in [0, \beta)$ such that $\beta = \sum_{k=1}^{|A|} \beta_k \times \frac{1}{|A|}$. We refer to this optimization problem as Minimum Activity-Specific Sensing ($MASS$).

Theorem 1. *$MASS$ is an NP-hard problem.*

Proof. In general, a non-convex non-linear optimization problem is a computationally hard problem. As shown in [10], such discrete optimization is a reduction of Subset Sum Problem which is known to be NP-complete. $MASS$ is also a discrete optimization problem with a non-convex non-linear constraint. Thus, the proof follows. □

The constraint in (4) functions as a black-box that outputs the performance drop in our multi-label classification system with respect to a given M. A multi-label classifier is a set of binary classifiers, usually formed as a hierarchical tree structure, where each internal node is a binary classifier that distinguishes two groups of labels and the leaves are single labels. We exploit this structure to approximate our optimization problem using dynamic programming. Dynamic programming is a widely used optimization technique that aims to solve the problem in a bottom-up manner by recursively solving the smaller but similar subproblems. Greedy approach is another popular technique for solving hard optimization problems by making locally optimal choices in each iteration with the hope of finding the global optimum. A polynomial time approximation algorithm can be found using both techniques. Greedy optimization is usually faster but it may not provide a good approximation. We use dynamic optimization by modeling the multi-label classification in form of a tree-based binary classification structure.

Definition 1 (Dynamic Optimization Tree)**.** A Dynamic Optimization Tree (DOT) is defined as a binary tree for multi-label classification where each internal node is a binary classifier/optimizer and each leaf represents a single class label. This structure is constructed, heuristically, using Algorithm 1.

There is an exponentially large number of ways for constructing such a binary tree. In fact, the number of possible

n-label classification trees is the $n\text{-}th$ Catalan number [11]. We use a heuristic approach to construct such a tree based on our similarity graph model.

Definition 2 (Complete Similarity Graph)**.** Given a multi-label classifier with $|A|$ labels where A is the set of all activities, we define a complete weighted undirected graph on $|A|$ nodes ($K_{|A|}$). Each vertex in this graph corresponds to one activity label (e.g., $v_i \leftrightarrow a_i$). The weight on each edge (ω_{ij}) connecting v_i and v_j vertices represents the similarity degree between the two activity labels, equivalent to ϵ_{ij}. The parameter ϵ_{ij} denotes the (i-j) pairwise classification error. Clearly, larger classification error means higher similarity degree between two classes. We refer to this graph as SG for short.

B. Problem Solution

The heuristic approach for finding the optimal dynamic optimization tree is based on the following observation.

Axiom 1. A higher compression ratio (i.e., less number of measurements) can be found when two signals with distinct labels are less similar.

The heuristic DOT construction approach aims to ensure that more similar classes have a higher-level lowest common ancestor. Intuitively, the subproblems (internal nodes in DOT) are preferred to minimize the measurements for dissimilar classes. Algorithm 1 describes the heuristic approach for finding the optimal DOT. Given the complete similarity graph, we iteratively break the graph (parent node) into two disjoint components (children nodes) until there is not any component in the complete similarity graph with an order larger than one (reaching the leaf nodes), resulting in a binary tree where its leaves are the vertices of the complete similarity graph. The heuristic resides on the method of cutting the graph into two disjoint components. Each possible s-t cut resulting in components S and T has a capacity defined by

$$c(S,T) = \sum_{(i,j)\in E_{SG}} \omega_{ij} d_{ij} \tag{5}$$

where E_{SG} is the set of edges in SG and d_{ij} is a binary variable equal to '1' if v_i and v_j belong to different components and equal to '0' otherwise. We need to find a cut that maximizes the difference between the two resulting components. In other words, given that ω_{ij} represent the similarity between two vertices, we need to find the s-t cut with minimum capacity. This is a global minimum-cut problem and can be solved in polynomial time using Stoer-Wagner algorithm[1]. We propose a polynomial time dynamic programming approach using the DOT structure. The dynamic optimization tree essentially breaks the initial optimization (i.e., $MASS$ problem) into branches of similar subproblems (i.e., internal nodes in DOT). For instance, consider a lowest-level internal node in DOT, with two leaves l_1 and l_2 (as its children) corresponding to activity labels a_1 and a_2, respectively. This internal node represents a subproblem which aims to minimize the measurement set $M = \{m_1, m_2\}$ satisfying a given threshold on the residual classification error (β). In a bottom-up fashion, higher-level internal nodes aim to find the minimum M using the solutions provided by their children. Each internal node finds the minimum measurement set M by finding a minimum of (1) the current solution of its children (i.e., M_l and M_r) and (2) $M_{(l)}^{\beta}$ subtracted by some $p \in \mathbb{N}$ and $M_r^{(\beta-\epsilon_{lp})}$ (or vice versa) where ϵ_{lp} is the residual classification error imposed by p. The solution given by the root node is the approximation to $MASS$ problem. The recursively computed solution in each internal node is given by

$$A(\beta,p) = \min\{||M^{(\beta)}||, A_l(\beta,p), A_r(\beta,p)\} \tag{6}$$

$$A_l(\beta,p) = ||\{M_l^{(\beta)} - p\} \cup M_r^{(\beta-\epsilon_{lp})}|| \tag{7}$$

$$A_r(\beta,p) = ||M_l^{(\beta-\epsilon_{rp})} \cup \{M_r^{(\beta)} - p\}|| \tag{8}$$

$$M^{(\beta)} = \arg_M \min_{q\in\{M_l \cup M_r\}} \{A(\beta,p)| \quad 0 < p < min(q)\} \tag{9}$$

where A, A_l, and A_r are the minimized M for the current node, the left child, and the right child, respectively. Set M is the measurement set corresponding to the minimum A, and also the output of the current node in the dynamic optimization tree. At any point, if $\epsilon > \beta$, $M^{(\beta-\epsilon)} = +\infty$. Algorithm 2 describes the proposed dynamic optimization.

Algorithm 1 *DOT* construction algorithm

```
procedure MAIN($SG$)
    Result: $DOT$   ▷ distance×similarity$^{-1}$ is maximized
    Input: $SG$   ▷ The complete similarity graph
    Initialization: $parent \leftarrow DOT.root;$
    Recursion:
     $(S,T) \leftarrow global\_min\_cut(SG);$
     For $G \in \{S,T\}$
        If $|V(G)| = 1$   ▷ G is a single node component
          $parent.add\_child(leaf, G);$ $Return;$
        $parent.add\_child(internal\_node, G);$
        Go to Recursion  $with\ SG \leftarrow G;$
```

Algorithm 2 Dynamic optimization approximation for $MASS$ problem

```
procedure MAIN($SG$)
    Result: $M$   ▷ minimzed $M = \{m_1, m_2, \ldots, m_{|A|}\}$
    Input: $DOT$   ▷ The dynamic optimization tree
    Initialization: $m_i \in M \leftarrow n;$   ▷ The Nyquist rate
    Tree traversal in reverse level order:
      Calculate $M^{(\beta)}$ from Equation (9);
    Return  $M^{(\beta)}$  $for\ root\ node;$
```

Theorem 2. *The proposed dynamic optimization approximation (i.e., Algorithm 2) satisfies the constraint in (2).*

Proof. In order to show that the residual classification error generated by the solution of Algorithm 2 is at most equal

[1] https://en.wikipedia.org/wiki/Stoer%E2%80%93Wagner_algorithm

to β, we just need to show that (6) holds the constraint in (2). It chooses the minimum between three elements. The first element, $M^{(\beta)}$, satisfies β, by definition. The second and third elements are similar, therefore we only examine A_l from (7). Subtracting M by p imposes a residual error ϵ_{lp}. As a result, in order to meet β, the solution of other child is recalculated using a new threshold $(\beta - \epsilon_{lp})$. Thus, the internal nodes (including the root node) preserves β and the result follows. □

Theorem 3. *The worst case time complexity of our approach is* $O(|A|^2N^2) + O(N|A|^2lg|A|) + O(n|A|^3N)$.

Proof. The complete similarity graph with size of $O(|A|^2)$ can be built in $O(|A|^2(N^2 + NlgN))$ using a decision tree classifier where $|A|$ is the size of the set of class labels and N denotes the size of dataset annotated with each activity label (refer to [12] for information on time complexity of training/testing a decision tree classifier). The binary tree construction is $O(|A|)$ if we access each node in $O(1)$ using a hash map data structure. Therefore the complexity of Algorithm 1, using StoerWagner algorithm in each iteration, will be $O(N \times |A|^2lg|A|)$. Using memoization technique[2], the time complexity of Algorithm 2 will be $O(n|A|((|A|N)^2+|A|Nlg(|A|N)))$ where n denotes the Nyquist rate. It accounts for the tree traversal, local minimization, and measuring classification errors. Therefore the overall complexity is $O(|A|^2N^2)$. The result follows. While in contrast, a brute-force solution grows exponentially with $o(n^{|A|}|A|N^2)$ time complexity which is impractical even for moderate-size inputs. □

C. Real-time Feedback

In the proposed adaptive compressed sensing framework, the sensing node autonomously generates the feedback in real-time without dependence on back-end process, as shown in Fig. 4b. It encompasses two modules: 'Interpolation' and 'coarse-grained activity recognition'. The compressed sensing module only captures a randomized fraction of samples (m/n). The interpolation module uses the simple yet effective piecewise linear interpolation in order to roughly estimates the underlying signal. The output of this module is then fed to the coarse-grained activity recognition module where the estimated signal is used to detect the type of activity with slightly lower confidence. The activity label predicted in this step is fed into the look-up table which is used to set the optimal measurement in 'adaptive compressed sensing' module. The look-up table is pre-set by the set of minimized activity-specific measurements (M) computed offline as explained previously.

IV. Evaluation

In this section we overview the experimental setting and the energy model used in our evaluations and then discuss the results achieved by the proposed design.

A. Experimental Setup and Energy Model

To demonstrate the effectiveness of the proposed framework, we use the real-world UCI 'Daily and Sports Activities Data Set' [13]. A 3D motion tracker placed on torso was used to collect triaxial acceleration data at $25Hz$ sampling rate. Eight subjects were asked to perform 19 activities (such as sitting, walking, running at different speed, cycling, playing basketball, etc.), each for duration of five minutes. The activity recognition process was identical in front- and back-end. We used a decision tree classifier for its simplicity and superior performance in activity recognition. Each trial was divided into five-second segments of activity where 20 statistical and morphological features were extracted from each segment. Refer to [14] for details on the feature set utilized in the experiments.

We define a power model for the proposed framework. The power consumption of such wearable sensor node consists of (1) data sampling (sensing cost); (2) on-node signal processing (including interpolation, feature extraction, and classification) and (4) data transmission to the base-station. Data sampling and transmission costs are calculated using the specifications of $ADXL335$ accelorometer[3] and ZigBee transmitter detailed in [15]. The accelerometer consumes $2.64\ mJ$ in active mode. Based on ZigBee protocol specifications, the cost of per byte communication is $1.49mJ$. In order to model the cost of on-node signal processing, we further calculated instruction level energy consumption of MSP430 micro-controller. The details of energy calculations is based on an instruction level energy model provided in [14]. The total energy consumption can be calculated by

$$Z(m) = m \times \lambda_s + (\frac{m}{5}) \times (\lambda_{pt}) \tag{10}$$

where m denotes the sensing rate, and λ_s and λ_pt represent the per-sample sensing cost, and per-segment processing and transmission costs.

B. Results

The initial activity recognition dataset is sampled at $25Hz$ meaning that each data segment contains 125 samples. The initial system setup achieves 88% accuracy using decision tree classifier and 10-fold cross-validation. We compare the results of the proposed adaptive sensing framework which utilizes the optimization algorithm discussed in Section III with the state-of-the-art nrmse-based minimization. While the optimization scheme proposed in this paper aims to minimize the number of measurements needed for each specific activity type by maintaining the overall classification accuracy over a given threshold, the nrmse-based approach minimizes m for each class just by constraining the nrmse of the signal reconstruction below an acceptable value (typically bellow 0.1). We also compare the performance of our system in terms of power saving against state-of-the-art approach [8].

One advantage of the proposed adaptive compressed sensing framework, is its ability to generate on-node feedback by

[2] https://en.wikipedia.org/wiki/Memoization

[3] https://www.sparkfun.com/datasheets/Components/SMD/adxl335.pdf

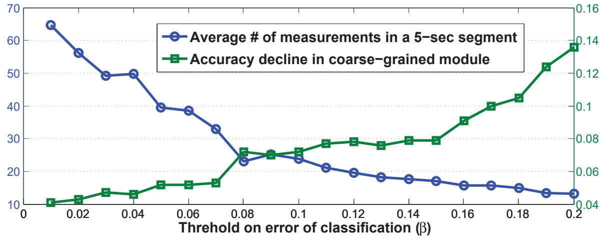


Fig. 5: average m per segment (blue) and accuracy decline when using coarse-grained activity recognition (green) shown for various thresholds.

TABLE I: Comparison of the proposed framework and state-of-the-art adaptive compressed sensing (conventional approach) in three cases where the gained activity recognition accuracy is 0.83, 0.86, and 0.87, respectively from top-down.

-	avg.	Energy/segment (mJ)				Saving vs
	m	Sens.	Proc.	Trans.	Total	initial (%)
Initial System	125	33.9	0	4.39	38.38	0
Conven. nrmse=0.1	62.6	17.1	0	2.2	19.22	49.9
Proposed ($\beta = 0.05$)	39.2	10.7	0.27	1.38	12.4	67.7
Conven. nrmse=0.05	97.6	26.5	0	3.43	29.97	21.9
Proposed ($\beta = 0.02$)	56.2	12.3	0.35	1.97	17.61	54.1
Conven. nrmse=0.01	122	33.1	0	4.29	37.46	2.4
Proposed ($\beta = 0.01$)	64.6	17.6	0.39	2.27	20.22	47.3

detection the activity type locally and without needing real-time connectivity to the back-end processing node. Fig. 5 shows the performance of the optimization scheme as well as the classification accuracy of the coarse-grained activity recognition module for various thresholds on the residual classification error (β). As you can see, when $\beta = 0.01$ the framework achieves 48% compression rate almost without loosing any performance. This number is 68.8% and 81.6% for $\beta = 0.05$ and $\beta = 0.08$, respectively. The difference in accuracy of the coarse-grained and fine-grained classifiers in shown in green. The significant performance of coarse-grained activity recognition module can be clearly observed, especially for smaller thresholds. For $0.01 < \beta < 0.08$, the lost in accuracy remained < 0.05.

Table I, compares the performance of proposed framework versus state-of-the-art approach as well as the initial system (traditional Nyquist sensing) in terms of the average number of measurements collected and energy consumption of various components associated with each data segment. The processing cost is associated with the on-node feedback generation (and, hence, is zero for other approaches) and is below $0.4mJ$/segment; which shows the ultra-low power functionality of the processing pipeline. Three values for nrmse is considered and the overall classification accuracy achieved using each one is used to set β in our approach. As a result, each pair of comparison is associated with the same classification performance. It shows our framework saves 17.8% to 45.1% more power compared to the nrmse-based approach.

V. Conclusion

In this paper, a novel activity-aware compressed sensing framework was proposed. Our approach has two key advantages over the state-of-the-art frameworks used in biosensing applications: (1) it uses an ultra-low power coarse-grained activity recognition module to generate feedback (i.e., the activity class label) at the fingertip of the sensing node without needing back-end communication; (2) it shifts the minimization paradigm from nrmse-based to performance based minimization by finding the optimal trade-off between overall activity recognition accuracy and per-class sensing rate. Our results showed superiority of the proposed design with high accuracy in feedback generation (less than 5% drop compared to fine-grained back-end reconstruction-classification) and up to 45% more power saving.

References

[1] M. Chan, D. Estève, J.-Y. Fourniols, C. Escriba, and E. Campo, “Smart wearable systems: Current status and future challenges,” *Artificial intelligence in medicine*, vol. 56, no. 3, pp. 137–156, 2012.

[2] A. Shahrokni, S. Mahmoudzadeh, R. Saeedi, and H. Ghasemzadeh, “Older people with access to hand-held devices: Who are they?” *Telemedicine and e-Health*, vol. 21, no. 7, pp. 550–556, 2015.

[3] S. A. Rokni and H. Ghasemzadeh, “Autonomous sensor-context learning in dynamic human-centered internet-of-things environments,” in *Proceedings of the 35th International Conference on Computer-Aided Design*. ACM, 2016, p. 75.

[4] J. Williamson, Q. Liu, F. Lu, W. Mohrman, K. Li, R. Dick, and L. Shang, “Data sensing and analysis: Challenges for wearables,” in *The 20th Asia and South Pacific Design Automation Conference*. IEEE, 2015, pp. 136–141.

[5] Y. C. Eldar and G. Kutyniok, *Compressed sensing: theory and applications*. Cambridge University Press, 2012.

[6] A. Wang, C. Song, X. Xu, F. Lin, Z. Jin, and W. Xu, “Selective and compressive sensing for energy-efficient implantable neural decoding,” in *Biomedical Circuits and Systems Conference (BioCAS), 2015 IEEE*. IEEE, 2015, pp. 1–4.

[7] D. Akimura, Y. Kawahara, and T. Asami, “Compressed sensing method for human activity sensing using mobile phone accelerometers,” in *Networked Sensing Systems (INSS), 2012 Ninth International Conference on*. IEEE, 2012, pp. 1–4.

[8] S. Yang and M. Gerla, “Energy-efficient accelerometer data transfer for human body movement studies,” in *Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC), 2010 IEEE International Conference on*. IEEE, 2010, pp. 304–311.

[9] B. K. Natarajan, “Sparse approximate solutions to linear systems,” *SIAM journal on computing*, vol. 24, no. 2, pp. 227–234, 1995.

[10] K. G. Murty and S. N. Kabadi, “Some np-complete problems in quadratic and nonlinear programming,” *Mathematical programming*, vol. 39, no. 2, pp. 117–129, 1987.

[11] G. D. Knott, “A numbering system for binary trees,” *Communications of the ACM*, vol. 20, no. 2, pp. 113–115, 1977.

[12] J. K. Martin and D. Hirschberg, “On the complexity of learning decision trees,” in *International Symposium on Artificial Intelligence and Mathematics*. Citeseer, 1996, pp. 112–115.

[13] B. Barshan and M. C. Yüksek, “Recognizing daily and sports activities in two open source machine learning environments using body-worn sensor units,” *The Computer Journal*, vol. 57, no. 11, pp. 1649–1667, 2014.

[14] H. Ghasemzadeh, N. Amini, R. Saeedi, and M. Sarrafzadeh, “Power-aware computing in wearable sensor networks: An optimal feature selection,” *IEEE Transactions on Mobile Computing*, vol. 14, no. 4, pp. 800–812, 2015.

[15] J. Baviskar, A. Mulla, M. Upadhye, J. Desai, and A. Bhovad, “Performance analysis of zigbee based real time home automation system,” in *Communication, Information & Computing Technology (ICCICT), 2015 International Conference on*. IEEE, 2015, pp. 1–6.