Power-Aware Activity Monitoring Using Distributed Wearable Sensors

Hassan Ghasemzadeh, *Member, IEEE*,
Pasquale Panuccio, *Student Member, IEEE*,
Simone Trovato, *Student Member, IEEE*,
Giancarlo Fortino, *Senior Member, IEEE*,
and Roozbeh Jafari, *Senior Member, IEEE*

***Abstract*—Monitoring human movements using wireless wearable sensors finds applications in a variety of domains including healthcare and wellness. In these systems, sensory devices are tightly integrated with the human body and infer status of the user through signal and information processing. Typically, highly accurate observations can be made at the cost of deploying a sufficiently large number of sensors, which in turn results in increased energy consumption of the system and reduced adherence to using the system. Therefore, optimizing power consumption of the system while maintaining acceptable accuracy plays a crucial role in realizing these stringent resource constraint systems. In this paper, we present an activity monitoring approach that minimizes power consumption of the system subject to a lower bound on the classification accuracy. The system utilizes computationally simple template-matching blocks that perform classifications on individual sensor nodes. The system further employs a boosting approach to enhance accuracy of the distributed classifier by selecting a subset of sensors optimized in terms of power consumption and capable of achieving a given lower bound accuracy criterion. A proof-of-concept evaluation with three participants performing 14 transitional actions was conducted, where collected signals were segmented and labeled manually for each action. The results indicated that the proposed approach provides more than a 65% reduction in the power consumption of the signal processing, while maintaining 80% sensitivity in classifying human movements.**

***Index Terms*—Action recognition, AdaBoost, distributed classification, low-power design, real-time embedded systems, signal processing, wearable computing.**

I. Introduction

WEARABLE medical devices, which utilize tiny and wearable sensor devices with embedded communication and processing units, offer unique opportunities for sensing, processing, and extraction of useful information from the human body. The main driving factors in designing this new generation of the healthcare paradigm include cost, power consumption, and wearability [1]. An important aspect of the low-power design is the development of efficient signal processing and data reduction algorithms that reduce computing load of the processing units, allowing for low-power low-cost processors.

Action recognition is a classification problem with the goal of detecting transitional actions (e.g., "Sit to Stand," "Walking," and "Kneeling") for a variety of applications such as physical activity monitoring [2], gait analysis [3], and diagnosis of many movement disorders [4]. Designing power-aware signal processing algorithms for action recognition is challenging due to maintaining acceptable classification accuracy while minimizing power consumption of the system [5].

The focus in this paper is on monitoring sparse movements as many monitoring applications only use a very small subset of events. For example, Parkinson's disease monitoring is only concerned with certain movements such as "Tremors" [6]. In real-time continuous patient monitoring, these target actions occur infrequently. Thus, the signal processing pipeline can be improved for specific detection of target actions. We use template matching to perform computationally simple classifications from a minimal set of body-worn sensors. We optimize this classification model for power consumption while accounting for system performance parameters such as precision and recall. Our power-aware boosting algorithm learns from a set of weak classifiers while minimizing system power consumption. The low-power signal processing is accomplished by using computationally simple classifiers that operate based on template matching on sampled inertial data and eliminating, from the processing pipeline, sensors not relevant to the activity monitoring. The high classification performance is maintained using a boosting approach that combines results from distributed sensors and enhances the classification accuracy by taking into account the contribution of individual sensors.

Novel aspects of our study are as follows: 1) We consider two performance criteria, accuracy and power consumption, to build a new classification model; 2) we use efficient template-matching blocks that are inherently less computationally intensive, use less memory, and are easy to implement; 3) we adapt an AdaBoost-driven combiner model to build a novel power-aware classifier learning approach that eliminates redundant sensors from the classification process; and 4) in contrast with studies that minimize the number of sensor nodes, our power-aware activity monitoring is a fine-grained sensor selection technique that eliminates inefficient sensors, selecting only those that are efficient in terms of power consumption and classification accuracy.

Our classification scheme relies on a template-matching block. Advantages of template-matching classification over classical classification algorithms that operate on feature space are as follows: 1) The template-matching function is easy to implement in either hardware or software as the implementations uses only Multiply-ACcumulator (MAC) operations; and 2) template-matching requires almost constant computation time.

II. Related Work

While the data collected by the sensor nodes can be processed in a distributed manner, most existing work focuses on developing algorithms for local processing of the data and using a data fusion scheme at the base station for summarizing state of the system. In a local processing paradigm, each sensor node performs partial processing on the data and transmits the results to a base station. The base station is responsible for combining data from all nodes and building a centralized classifier that identifies unknown actions. The accuracy of such a classifier depends on a variety of parameters including the classification algorithm, sensor node placement, types of features extracted from the signal, and number and type of actions/activities to be recognized. While many algorithms such as k-nearest neighbor (k-NN) [7], hidden Markov models (HMM) [8], Naive Bayes [9], and support vector machines [10] have been investigated, the k-NN and HMM are more common in action recognition in the wireless healthcare domain when motion sensors are generally used for information inference.

Logan *et al.* [11] present a study on activity recognition using different types of sensory devices, including built-in wired sensors, RFID tags, and wireless inertial sensors. The analysis performed on 104 h of

Manuscript received June 23, 2013; revised March 19, 2014; accepted March 20, 2014. Date of publication June 4, 2014; date of current version July 11, 2014. This paper was recommended by Associate Editor Z.-H. Mao.

H. Ghasemzadeh is with the School of Electrical Engineering and Computer Science, Washington State University, Pullman, WA 99164 USA (e-mail: hassan@eecs.wsu.edu).

P. Panuccio, S. Trovato, and G. Fortino are with the Dipartimento di Informatica, Elettronica e Sistemistica, Universitá della Calabria, 87036 Rende, Italy (e-mail: pasquy@i2000net.it; sim.trov@gmail.com; g.fortino@unical.it).

R. Jafari is with the Department of Electrical Engineering, University of Texas at Dallas, Richardson, TX 75080 USA (e-mail:rjafari@utdallas.edu).

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

Digital Object Identifier 10.1109/THMS.2014.2320277

data collected from more than 900 sensor inputs shows that motion sensors outperform the other sensors on many of the movements studied. A prototype called *MEDIC*, developed in [12] for remote healthcare monitoring, uses a personal digital assistant as the base station and several sensor nodes that collect and process physiological data. A Naive Bayes classifier [13] provides more than 90% accuracy. A wireless body sensor system for monitoring human activities and location in indoor environments is introduced in [14], where each sensor node is equipped with an accelerometer, gyroscope, and magnetometer. Bao *et al.* [15] use a network of five accelerometers to classify a sequence of daily activities. They report a classification accuracy of 84% for detecting 20 actions. The system in [16] uses seven different sensors embedded in a single node to classify 12 movements and achieves a classification accuracy of 90%. The accuracy reported by the centralized k-NN classifiers in [17] is greater than 90% for classification of different human actions.

Reducing the number of active nodes is a common approach for power optimization and wearability enhancement in body sensor networks. Ghasemzadeh *et al.* [17] formulate the coverage problem in the context of movement monitoring using inertial on-body sensors. Their technique focuses on the minimum number of sensor nodes that produces a full action coverage set. Another way to reduce the number of active nodes is to keep track of the actions performed and to use the subset of sensors that can observe transitions from the current motion [18]. Zappi *et al.* [19] propose to optimize the system energy consumption by selecting the required subset of sensors with meta-classifier sensor fusion. Therefore, it is sufficient to turn ON sensors only when their values are needed to enhance accuracy. In [20], only a few sensor nodes are required to correctly classify a small action set. Additional sensor nodes are added to the system, if new actions cannot be uniquely identified by the existing sensors. Lombriser *et al.* [21] introduces a model for dynamic reconfiguration of the system, thus reducing the number of active nodes.

The idea of combining simple classifiers to achieve higher accuracy results is discussed in [22], where authors suggest using a single accelerometer for activity monitoring and combining classifiers using plurality voting. A lot of work has been done on selecting only relevant pieces of information for classification. Selecting only individual features for each activity can improve the rate of the classification as in [23]. In [24], AdaBoost is used to select a small number of features to speed classification. The algorithm automatically selects the best features and ranks them based on their classification performance. Given the maximum number of features that the activity recognition system can use, the system automatically chooses the most discriminative subset of features and uses them to learn an ensemble of discriminative static classifiers for the activities that need to be recognized.

III. Activity Monitoring Model

In this section, we present the architecture of our activity monitoring model (see Fig. 1), which uses template matching for local classification and a weighted combiner model for global decision making. In Section IV., we will use this action recognition model to formulate a problem for sensor selection.

Our signal processing algorithms use a simple template-matching model to reduce complexity and customize the processing model for action recognition applications [25]. Our approach is based on normalized cross correlation (NCC) [26] which is lightweight, fast, and adaptable for real-time implementations. Each sensor stream is associated with a binary classifier that runs on the sensor node. The classifier performs coarse analysis on the motion signals by comparing the incoming signal with a predefined template associated with the target

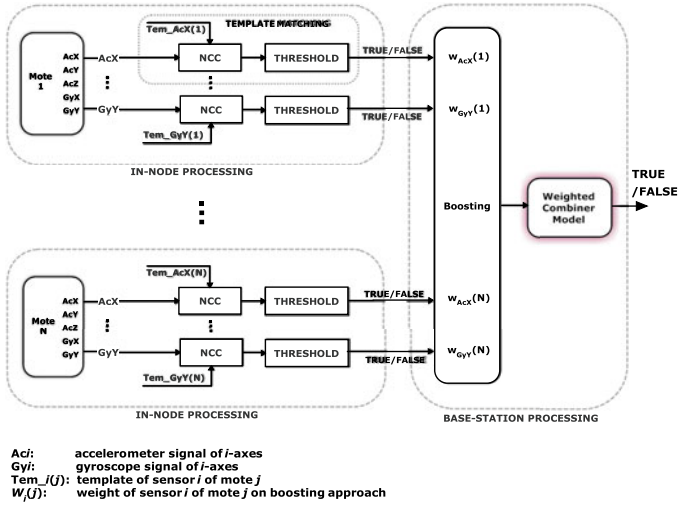

Fig. 1. Template-matching and boosting approaches for action recognition.

action. The template matching generates a similarity score between the signal and the target template. The calculated score is compared against a threshold in order to classify the incoming signal into a binary truth value (true if the current action is classified as a target action and false for a nontarget action).

The model needs only the template of the target action to be stored on the sensor node, and it requires only the computation of the NCC score (which is faster than other similarity measures such as dynamic time warping and HMM approaches typically used in activity recognition). We design a "weak classifier" for each sensor on the node and combine local decisions to create a significantly stronger classifier at the base station. Per-node weak classifiers support implementation of the signal processing algorithms in the node [27], [28]. In addition, the classifier combiner ensures improved accuracy. Since the signal processing is done on the node, the amount of data sent to the base station is reduced, resulting in a reduction of energy consumption and bandwidth.

A. Template Generation

To obtain a unique template for the target action, we use a supervised learning approach. This task is accomplished during training and is independent of the real-time constraints of the system. The raw sensor readings are collected along with the video recording of the actions. The collected signals are then segmented and labeled manually for each action. Video recording is used to segment the data in a more fine-grained manner. This manual segmentation ensures precise segmentation and labeling of the data and limits injecting automatic segmentation errors to subsequent signal processing and pattern recognition blocks. The action template needs to be representative of the specific action while being robust to variabilities between different people and movement instances. Our templates are generated by comparing every pair of training instances and choosing the one which is most similar to the others. The NCC measure is used to calculate similarity score between pairs of training instances.

Action instances may vary in length. While a template has a fixed length, the incoming signal segment may have a different length. To address this problem, we assume that the template and the incoming signal segment have the same length. A normalization technique is used to make the two signals uniform in length. The normalization process

for two signals S and T is given by

$$newS(i) = S\left(\left\lfloor \frac{|S|}{|T|} \right\rfloor \times i\right) \quad \forall i \in \{1, 2, \ldots, |T|\} \tag{1}$$

where T is the longer signal and $newS$ is the transformation of the shorter signal.

B. Weak Classifier

Each sensor node in our system is composed of several weak classifiers, each associated with a physical sensor such as *x*-accelerometer, *y*-accelerometer, and *z*-accelerometer. The first step in our model is an in-node classification of the signals being generated by the inertial sensors using weak classifiers. This classification is accomplished using a threshold value for each axis of the sensors. The threshold is calculated to distinguish between target and nontarget instances. The value for each weak classifier is calculated during training and is set to fall between upper and lower bounds. These bounds are the average of the cross-correlation values by comparing the template with positive and negative instance classes.

Each weak classifier makes a decision on the incoming signal as follows. For each axis of the sensors (e.g., three for accelerometer and two for gyroscope), the corresponding signal is used to perform the template matching and compare the incoming signal with the previously determined template. If the result of the comparison is greater than the threshold, the signal is labeled as a target action. Otherwise, the signal is classified as a nontarget action.

C. Weighted Combiner

Although in-node classifiers are computationally simple and fast, their accuracy is limited by the relatively small amount of information they observe from human actions. In fact, each weak classifier uses only a 1-D signal to perform template matching. In order to enhance interclassifier performance, we employ a decision fusion approach based on ensemble learning [29]. Our decision fusion algorithm is a classifier combiner, which takes into account the amount of contribution each weak classifier can make to the overall accuracy of the system.

Although our decision fusion approach is inspired by the ensemble learning, it is quite different from classical techniques. Traditional ensemble learning techniques iteratively resample the training set and train the classifier with a new resampled training set at each phase of the algorithm. In contrast, we work with the original training instances without resampling them, and a weight is assigned to one of the classifiers at each iteration of the algorithm. Our basic boosting approach is presented in Algorithm 1.

Algorithm 1: Weighted Combiner Model

data: TS training set $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$,
L learning algorithm generating hypothesis $h_t(x)$ classifier,
N size of training set
initialize: d_n weight of instance n (d is a distribution
with1 $= \sum_{n=1}^{N} d_n$)$d_n = 1/N$
input : T max number of hypotheses in the ensemble
begin
  for $t = 1, \ldots, |T|$ **do**
    1. Train weak learner and obtain hypothesis h_t
    2. Compute error $\epsilon_t = \sum_{n=1}^{N} d_n I(y_n \neq h_t(x_n))$
    3. Compute hypothesis weight $\alpha_t = \frac{1}{2} \ln \frac{1-\epsilon_t}{\epsilon_t}$

output: combined hypothesis $f_{Com}(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$

The algorithm works with a set of training instances that are correctly labeled with the appropriate actions. Each instance consists of a sensor reading (i.e., x_i) and a label (i.e., y_i) and is associated with a weight that is equal for all instances of the same action. Let L be the learning algorithm that generates the hypothesis $h_t(x)$. The algorithm L is the method of finding the threshold, and $h_t(x)$ is the function of our weak classifier that produces true/false labels. If the outcome of the comparison between instance x and target action template is greater than or equal to the threshold value, a true label is generated. Each weak classifier is associated with a weight, depending on the classification error that is obtained during training. The weak classifiers that achieve a smaller error are assigned larger weights to signify their contribution to the overall system performance.

IV. Power Optimization Approach

Classical AdaBoost learns from a set of weak classifiers and boosts classification performance by allocating weights to the individual local classifiers. The weights specify contribution of each classifier to the overall classification task. There are two main drawbacks with this approach.

1) AdaBoost examines all classifiers even if they provide less informative data for classification. Contribution of different sensors to action recognition varies depending on the placement of the sensor, type of inertial information, and actions of interest. Practically, body-worn sensors can produce redundant or correlated information when a movement occurs. For instance, signal values from nodes placed on the "upper arm" and "lower forearm" correlate during an upper body movement. In addition, data provided by a node placed on the "leg" may not contribute to upper body movements. Consequently, a more intelligent learning algorithm could select only those sensors that better contribute to the classification accuracy.
2) AdaBoost does not take into account the power requirements of the individual weak classifiers while learning from weak classifiers. In fact, the weights assigned by the AdaBoost account only for the accuracy of the classifiers. However, power consumption of the classifiers varies depending on the type of sensors, computing complexity, and data communication requirements. For example, gyroscopes generally consume much more power than accelerometers. Therefore, the optimal subset of the classifiers is selected by making an appropriate design tradeoff between accuracy and power consumption.

A. Problem Formulation

Our system aims to detect a target action of interest, $\hat{a}$, associated with a particular template. Therefore, the system consists of several binary weak classifiers each contributing to detection of the target action using a template-matching approach. To build a classification framework, we assume that there are a total of m nontarget actions, i.e., $A = \{a_1, a_2, \ldots, a_m\}$, that may occur during the operation of the system.

A motion sensor node, s_i, is a physical wearable node composed of l sensors that capture inertial data from human movements. Let $S=\{s_1, s_2, \ldots, s_n\}$ be a set of n sensor nodes that are used to distinguish between the target action and nontarget actions. An example of a sensor node used for our proof-of-concept evaluation has $l = 5$ sensors including three axes of accelerometer and two axes of gyroscope readings.

Each sensor node, s_i, consists of l weak classifiers: $C_i = \{C_{i1}, C_{i2}, \ldots, C_{il}\}$. Each classifier C_{ik} is associated with one of the

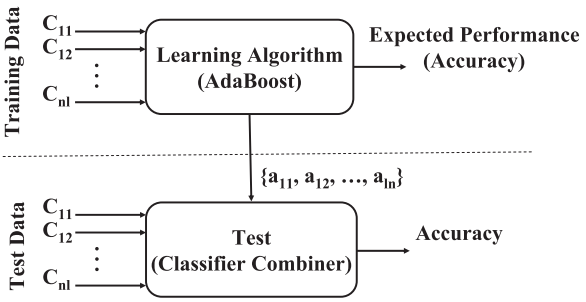

Fig. 2. Learning algorithm and classifier combiner during training and test.

sensors available on s_i and is a binary classifier that makes a classification decision using the similarity score obtained from NCC.

Given the n nodes and l sensors per node, the system has a total of $T = n \times l$ weak classifiers. Fig. 2 shows how learning parameters (e.g., weights $\alpha_{11}, \ldots, \alpha_{nl}$) are generated during training. The learning algorithm can also provide an estimate of the expected accuracy of the entire classification, α. We note that the performance modeling in Fig. 2 relies solely on the training data used in the learning phase and is not biased by any close feedback input from the test module.

To calculate power consumption of the set of active classifiers, we consider two sources of power consumption, namely computation and communication costs. We assume that each classifier is associated with a specific sensor (e.g., x-accelerometer, y-gyroscope) and, therefore, has a fixed computation cost depending on whether or not it is selected.

Definition 1 (Computation Cost): For each classifier C_{ij} associated with ith node and jth classifier, we define w_{ij} as the computation cost associated with power consumption of the corresponding sensor. This value is *a priori* known and has a nonzero value for active classifiers, while it is zero for nonactive sensors

$$P_{\text{comp}} = \sum_{i=1}^{n} \sum_{j=1}^{l} x_{ij} w_{ij} \tag{2}$$

where x_{ij} denotes if classifier C_{ij} is active

$$x_{ij} = \begin{cases} 1, & \text{if classifier } C_{ij} \text{ is active} \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Definition 2 (Communication Cost). For a set of weak classifiers used for learning, the communication cost is given by

$$P_{\text{comm}} = \sum_{i=1}^{n} f\left(\sum_{j=1}^{l} x_{ij} b_{ij} \right) \tag{4}$$

where $f(.)$ denotes the communication cost due to transmission of certain amount of data, x_{ij} denotes if classifier C_{ij} is selected, and b_{ij} represents the amount of data that is generated by classifier C_{ij} and needs to be transmitted to the base station.

We note that the communication costs are calculated for each sensor node rather than individual sensors/classifiers. This is mainly due to combining results of all active classifiers at each node prior to transmission to the base station. In other words, energy cost of communications is calculated collectively for all active sensors at each node.

The power consumption of the system due to selection of a set of weak classifiers used for learning is then given by

$$Z = P_{\text{comp}} + P_{\text{comm}}. \tag{5}$$

Problem 1: Given a set of $T = \bigcup C_i = \{C_{11}, C_{12}, \ldots, C_{nl}\}$ classifiers, as well as data units b_{ij} and computation cost w_{ij} for each classifier C_{ij}, the problem of minimum cost classifier selection (MCCS) is to find a subset of C_{ij} with minimum total cost while a lower bound of $\alpha \geq F$ on the overall accuracy is met.

Therefore, the optimization problem can be written as follows:

$$\text{Minimize} \sum_{i=1}^{n} \sum_{j=1}^{l} x_{ij} w_{ij} + \sum_{i=1}^{n} f\left(\sum_{j=1}^{l} x_{ij} b_{ij} \right) \tag{6}$$

subject to

$$\alpha \geq F. \tag{7}$$

B. Problem Complexity

The optimization problem presented in (6) poses several difficulties in arriving at an optimal solution: 1) The communication cost in (4) is a concave function, and minimizing a concave function is considered to be hard in general [30]. Concavity of the communication cost is due to the decrease in energy per bit as packet size increases. In fact, larger packets consume less energy per bit, turning the cost function into a concave function [31]. 2) The constraints of the optimization problem in (7) are nonlinear inequalities, which makes the minimization problem even harder. Therefore, in its general case, the MCCS problem belongs to the broad class of concave minimization problems over nonlinear constraints.

To develop a polynomial-time algorithm for the MCCS problem, we simplify some assumptions made for our classification model. Transforming the concave communication cost function into a linear function of the transmitted data turns the objective function into a convex function. One specific property of our weak classifiers is that they produce a fixed (e.g., λ) and small amount of data per classification. That is, upon occurrence of a new action, each active classifier generates a label (true/false) as (see Fig. 1). Typically, only a few bits are generated by each sensor node depending on the number of active sensors. When the number of sensors embedded within each node is small, the communication costs are almost constant for the nodes with at least one active sensor. In other words, for each sensor/classifier, we have

$$f(x_{ij} b_{ij}) = \begin{cases} \lambda, & \text{if classifier } C_{ij} \text{ is active} \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

For example, the amount of energy per bit is about 0.67 μJ when data transmission rate is 1 bit/s. Increase in the amount of energy per bit is negligible when the data rate increases by a few bits per second, mainly because the packet header is the dominant factor in the packet size. Thus, it is reasonable to assume that

$$f\left(\sum_{j=1}^{l} x_{ij} b_{ij} \right) = f(x_{ij} b_{ij}) = \lambda. \tag{9}$$

Therefore, local results can be combined to form a small fixed-size packet. As a result, it is reasonable to assume that the communication cost is constant for each active classifier. With this, the objective function in our optimization problem can be rewritten as follows:

$$\text{Minimize} \sum_{i=1}^{n} \sum_{j=1}^{l} x_{ij} (w_{ij} + \lambda) \tag{10}$$

where λ is the communication cost for each active node. This would turn our optimization problem into minimizing the overall computation cost.

C. Greedy Approach

In this section, we present a greedy heuristic algorithm that takes power consumption and accuracy of the weak classifiers into account and finds a subset of the least power-consuming classifiers that maintain an overall accuracy of at least equal to a given value. Our greedy algorithm is a backward elimination algorithm that starts with all classifiers being considered for AdaBoost learning. The set of all classifiers can potentially achieve a maximum accuracy that might be much higher than the lower bound accuracy. However, this configuration is highly power consuming as it uses all available sensors for action recognition. The algorithm iterates through different steps until it stops given a condition on overall accuracy of the classification. At each stage of the algorithm, one of the classifiers is eliminated. The elimination criterion is validated as a tradeoff between power consumption and accuracy. Specifically, the classifier whose $\frac{w_{ij}}{\alpha_{ij}}$ is maximized is the candidate for elimination. We note that w_{ij} and α_{ij} represent the power consumption and significance of the classifier C_{ij}, respectively. Yet, at each step, the algorithm checks whether the overall accuracy is still above the minimum desirable value. The algorithm stops if elimination of the next classifier would decrease the overall accuracy to a value below the given desirable threshold, F.

V. Proof-of-Concept Evaluation

In this section, we present a proof-of-concept evaluation of our classification framework and power-aware sensor selection algorithm. The data were collected from three males in good health. The data include acceleration and angular velocity for these 12 transitional actions: 1) stand to sit, 2) sit to stand, 3) sit to lie, 4) lie to sit, 5) bend and grasp, 6) kneeling, right leg first (Kneel-R), 7) kneeling, left leg first (Kneel-L), 8) turn clockwise 90° (CW90), 9) turn counterclockwise 90° (CCW90), 10) move forward—1 step (Move F), 11) move backward—1 step (Move B), and 12) jump. Each participant was asked to perform each one of the actions ten times while wearing four motion sensors on these body locations: 1) right wrist, 2) waist, 3) right thigh, 4) right ankle. Each wearable sensor node had five sensors including three axes of accelerometer and two axes of gyroscope. The data collected from the experiments were stored on a Laptop computer where we developed our signal processing algorithms and conducted the analytic results.

A. Training Parameters

A set of experiments was conducted to identify the 12 actions listed previously. For each experiment, the particular action was considered as target action, and the rest of the actions formed a nontarget class. As discussed previously, for each target action, we generate a template that will be used during training, for parameter setting and threshold calculation, and during test to compare the template against the incoming action. The training dataset for each target action was used to generate the corresponding template for each weak classifier. We recall that each classifier is associated with one axis of the inertial sensors. The training instance with the minimum average distance from all others was considered as the template of the target action.

Once a target template is determined, a threshold value needs to be calculated for each weak classifier. The threshold will be used later during classification to accept or reject an incoming action as the target action. Since NCC is used to compare the signals, the range of the cross-correlation scores resulting from the comparison varies between -1 (uncorrelated) and $+1$ (correlated). To detect the threshold, we compared the template with the "true" and "false" instances. The true instances were the sensor signals referring to the target action and the false one to the nontarget ones. Comparing the template with the true instances would output values close to $+1$ as they refer to the same action, whereas comparing the false instances will produce lower correlation values.

By taking an average over all the similarity scores between the template and target action instances, we obtained an upper bound on the value of the threshold as given by

$$\text{Thr}_{\text{upper}} = \frac{1}{N}\sum_{i=1}^{N}\gamma(\text{TPL}, S_i) \quad \forall S_i \in \text{TrueClass} \tag{11}$$

and similarly, a lower bound on the threshold value was calculated by comparing all nontarget action trials with the target template:

$$\text{Thr}_{\text{lower}} = \frac{1}{N}\sum_{i=1}^{N}\gamma(\text{TPL}, S_i) \quad \forall S_i \in \text{FalseClass}. \tag{12}$$

Thus, the value of the threshold can vary between $\text{Thr}_{\text{lower}}$ and $\text{Thr}_{\text{upper}}$, changing the result of the classification in terms of number of false positive and false negative values. For our experiments, we use a threshold that is calculated as given by

$$\text{Thr} = \frac{\text{Thr}_{\text{upper}} + \text{Thr}_{\text{lower}}}{3} \tag{13}$$

which was chosen empirically.

B. Detecting "Sit to Stand"

In our first analysis, we trained the system for detecting one particular action (i.e., "sit to stand") as the target action. The greedy algorithm was used to find the best subset of the sensors that can detect this action with a precision of $P \geq 90\%$ and a recall of $R \geq 80\%$. The backward elimination algorithm removed all weak classifiers except one sensor axis. The resulting active sensor was the "*Z*-axis accelerometer" of the "right thigh" node. This observation can be interpreted as follows. The *Z*-axis acceleration is the axis for the frontal plane of the participant and has a unique pattern during "sit to stand." This pattern is not repeated for the rest of actions and, therefore, is a distinguishing pattern that can by itself distinguish between this action and the others. This results in 20-fold reduction in the number of active sensors (from 20 to 1).

C. Accuracy and Power Analysis for All Actions

The system may be used to detect any target action. Thus, it is important to measure performance for actions other than when "sit to stand" is the target event. For this reason, we performed an analysis in this section where each action is considered as the target, and performance of the system is analyzed accordingly. We analyzed these results for three scenarios where performance of the desired classification performance varies. For each performance level, we found the minimum set of weak classifiers that achieve the given sensitivity (or recall) and precision values. The three performance levels are:

1) First level: precision $= 0.8$, recall $= 0.7$.
2) Second level: precision $= 0.85$, recal $= 0.75$.
3) Third level: precision $= 0.9$, recall $= 0.8$.

1) Classifier Accuracy Performance Analysis: Table I shows the number of sensors (or weak classifiers) that are required to obtain the desirable performance. These classifiers are needed to achieve a precision and a recall that is equal or greater than the given performance level. Of course, to increase the accuracy, for certain actions, we might

TABLE I
LIST OF ACTIVE SENSORS (WEAK CLASSIFIERS) FOR DETECTING EACH TARGET ACTION

Movement	Desired accuracy threshold values (P = Precision, R = Recall, wk = weak classifier)					
	P = 0.8 R = 0.7	**No. of wk**	**P = 0.85 R = 0.75**	**No. of wk**	**P = 0.9 R = 0.8**	**No. of wk**
Stand to Sit	AcZ-3	1	AcZ-3	1	AcZ-3	1
Sit to Stand	AcZ-3	1	AcZ-3	1	AcZ-3	1
Sit to Lie	AcZ-4	1	AcZ-4	1	AcX-1,AcY-2,AcZ-4	3
Lie to Sit	AcY-4	1	AcY-2, AcZ-3, AcY-4	3	AcY-2, AcZ-3, AcY-4	3
Bend Grasp	AcX-1, AcZ-2, AcZ-3	3	AcX-1, AcZ-2, AcZ-3	3	AcX-1, AcY-1, AcY-2 AcZ-2, AcZ-3	5
Kneel-R		15		20		20
Kneel-L	AcX-1, AcZ-1, AcY-4	3	AcX-1, AcZ-1, AcY-4	3	AcX-1, AcZ-1, AcY-4	3
CW90		14		14		17
CCW90	AcX-2, AcY-3, AcZ-3 AcX-4, AcY-4	5	AcX-2, AcY-3, AcZ-3 AcX-4, AcY-4	5		10
Move F	AcZ-1, AcZ-2, AcY-3	3	AcZ-1, AcZ-2, AcY-3	3	AcZ-1, AcZ-2, AcY-3	3
Move B		20		20		20
Jump		14		14		20

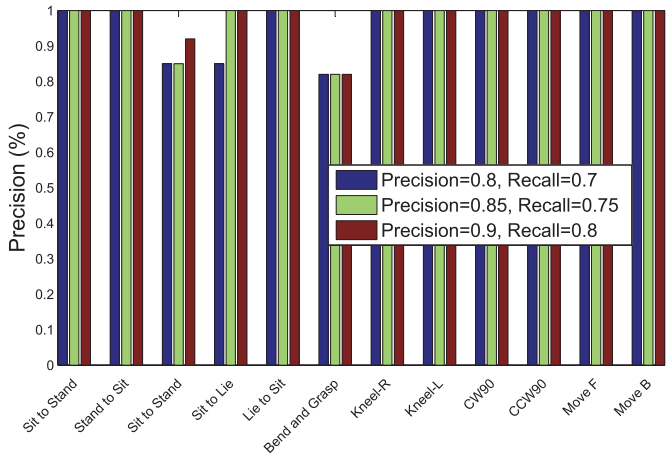

Fig. 3. Measured precision due to using only active sensors for classification. Results are presented for three different performance levels.

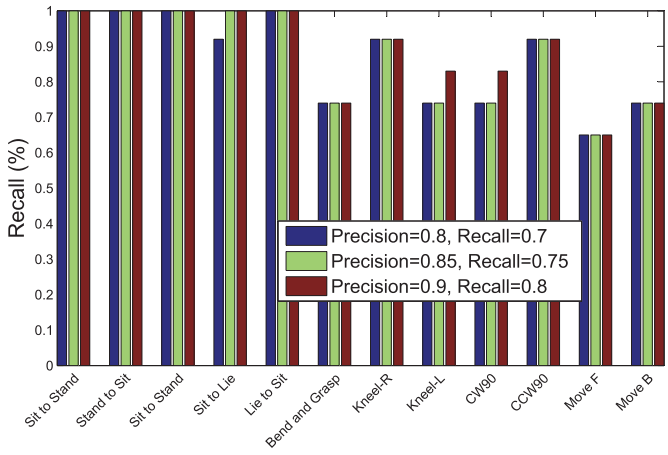

Fig. 4. Measured recall by selecting active sensors reported by backward elimination greedy algorithm.

need to increase the number of weak classifiers that should be used by the final classifier combiner. The number of active sensors clearly depends on the action that is intended to be recognized. In most cases, only a few sensors are enough to monitor the target action. However, there are cases such as actions Kneel-R, CW90, Move B, and Jump where a larger number of sensors need to be selected during classification. The list of the weak classifiers used during the classification task, for different levels of accuracy, is shown in Table I. For visualization, the list of the active sensors is eliminated from the table for those actions that require a large number of active sensors. This is the case for actions Kneel-R, CW90, Move B, and Jump, which require 15, 17, 10, and 14 active sensors for the highest calculated performance, respectively. Figs. 3 and 4 illustrate the results of the classification in terms of measured precision and recall. These results are based on the minimum number of weak classifiers.

2) Power Analysis: Reducing power consumption of the system usually results in a decrease in classification accuracy of the system. The model developed in this paper aims to minimize power consumption of the system while taking into account classification performance that can be obtained from signal processing and pattern recognition algorithms. As mentioned previously in Section IV, we assume that power consumption due to data transmissions is constant for each active sensor node, and therefore, only power consumption due to template matching and local classifications specifies the most power-consuming weak classifier at each state of the greedy algorithm. The nominal values of power used by the accelerometers and the gyroscopes on TelosB mote are 2.64 and 31.35 mW, respectively. Given that data acquisition from a sensor (i.e., accelerometer or gyroscope) is constant regardless of the number of axes used for classification, we allocate a fixed cost to each sensor even if only a single weak classifier is used. As a result, the power consumption of each weak classifier is equal to the power consumption of the corresponding sensor axis.

In Table II, the power consumption of the classification for each target action is shown for the case of highest precision and recall. As we can see, the values of the power consuming remain very low (e.g., below 10 mW) for almost all the actions. The power needed for the classification increases suddenly when we start to use the weak classifiers associated with the gyroscope sensors. Such power-consuming

TABLE II
POWER CONSUMPTION DUE TO POWER-AWARE LEARNING

Mov.	W/ Optimization [mW]	W/o Optimization [mW]	Saving [%]
Stand to Sit	2.6	135.9	98.1
Sit to Stand	2.6	135.9	98.1
Sit to Lie	7.9	135.9	94.2
Lie to Sit	7.9	135.9	94.2
Bend Grasp	7.9	135.9	94.2
Kneel-R	135.9	135.9	0
Kneel-L	5.3	135.9	96.1
CW90	104.6	135.9	23.0
CCW90	10.6	135.9	92.2
Move F	7.9	135.9	94.2
Move B	135.9	135.9	0
Jump	135.9	135.9	0
Avg.	47.1	135.9	65.4

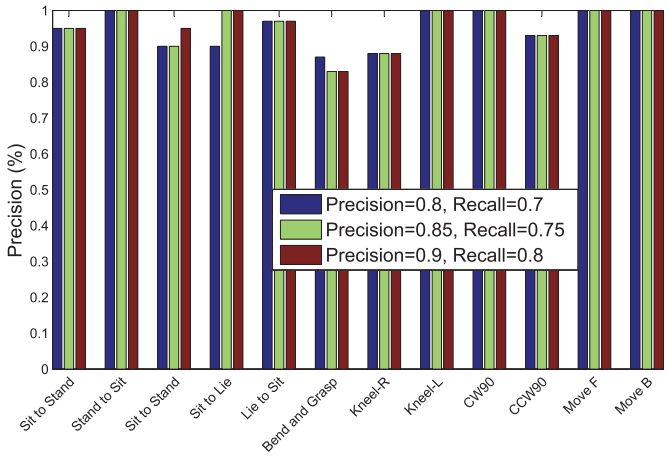

Fig. 5. Cross-validation analysis of precision with set of minimum sensors for classification.

classifiers are selected in order to improve the accuracy of the classification and to achieve the desirable classification performance as requested by the user. The amount of power savings achieved by using our power-aware action recognition framework is shown in the last column in Table II.

D. Generalizability

The results that we presented in Section V-C are based on a fixed training/test set and using a template-matching approach that we have developed in this paper. We chose a fixed training/test set (rather than a cross-validation approach) in order to be able to demonstrate active sensors, as shown in Table I. In this section, we demonstrate how our results can be generalized under a cross-validation scenario. Furthermore, we show accuracy performance comparison between our approach and a k-NN classifier.

Our performance analysis of template matching relies on two single sets: training and test. We want to analyze our combiner classifier using k-fold cross-validation approach, which is powerful in estimating more accurately the classification performance of our system when training and test sets change. Figs. 5 and 6 show the measured precision and recall for the classifier, using $k = 10$ folds and stratified random subsampling for their construction.

We compared the template-matching results against a feature-based classification approach, often used in activity recognition context: k-NN. In this approach, each sensor calculates some features on a window

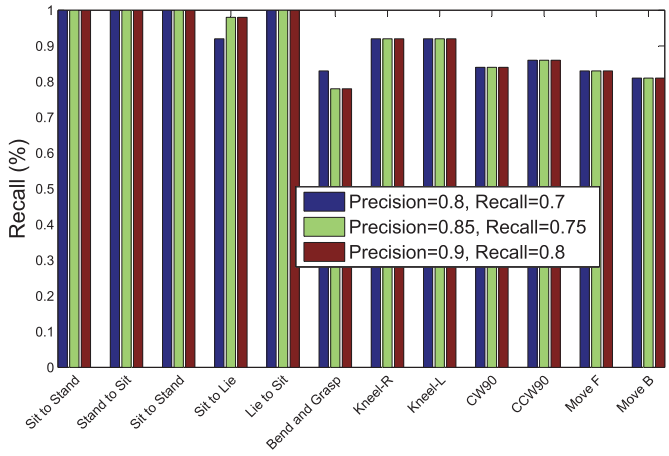

Fig. 6. Cross-validation analysis of recall with minimum sensors set for classification.

TABLE III
COMPARISON OF OVERALL PRECISION AND RECALL LEVELS

Approach	Precision [%]	Recall [%]
Template Matching	97.8	88.9
k-NN	89.2	83.3
Cross Validation	96.5	91.3

of acquired raw data; then, each set of features is sent to the base station, where the classifier labels the class by a majority vote among the k closest training examples. Distance between the data is evaluated according to a metric. The most commonly used metrics are Euclidean distance, Manhattan distance, and Hamming distance. Features extracted for classification are basically statistical information about the signal such as mean, standard deviation, energy, and correlation.

We simulated the classification approach in MATLAB, using the k-NN classifier with $k = 5$ (a reasonable value for reducing classification noise) and Euclidean distance metric; for our test purposes, sensors extracted four features from the raw data including mean, standard deviation, minimum value of signal segment, and maximum amplitude of the signal segment. We evaluated the approach using the same dataset used for template matching and, for each movement, only the selected sensors resulting by backward elimination algorithm. We show the average precision and recall values among the movements, concerning only the third performance level and for different test scenarios including template matching with fixed training/test sets, k-NN classification, and cross-validation approach.

Results in Table III show that, using the same set of sensors for classification, template matching has a generally better performance level than k-NN: 8.64% for precision and 5.56% for recall. Cross validation also demonstrates our classifier effectiveness, showing an excellent generalization level of template matching: precision and recall reach a value higher than 90%.

VI. CONCLUSION

In this paper, we proposed a novel classification approach that incorporates two design criteria: energy consumption and classification accuracy. Our system uses simple template-matching blocks to perform coarse classification of human movements on wearable sensor nodes. Only a subset of the sensors is selected for classification purposes where active sensors are determined according to their power consumption as

well as their contribution to the overall classification performance of the system. The results obtained by active sensors are further combined through a boosting approach to achieve higher classification accuracy.

As discussed in Section V-D, our template-matching algorithm outperforms the k-NN classifier by achieving 8.64% higher precision and 5.56% higher recall values. Our approach achieves both precision and recall values of more than 90%. Accuracy performance of the proposed classification algorithm can also be compared against a number of previously studied approaches. The Naive Bayes classifier presented in [13] achieves 90% accuracy, but exact precision and recall values are not reported. The classification accuracy reported in [15] is 84% for detecting 20 movements using a network of five motion sensor nodes. The accuracy obtained in [16] is 90% for classifying 12 movements using seven different sensors embedded in a single node.

Our template-matching classification approach has several advantages over classical algorithms:

1) the template-matching function is easy to implement in either hardware or software as it only uses MAC operation;
2) template matching requires almost constant computation time.

Despite these advantages, template matching may not be feasible for all physical movement monitoring applications. While further in-depth study is needed to draw solid conclusions, we think that major limitations of the proposed approach fall into the following areas: 1) While our experimental analysis shows that template matching is effective in detecting typical transitional movements, simple template matching may fail when dealing with more complex movements. For example, the type of movements studied in [21] may require a different classification scheme or a hierarchy of template-matching blocks for reliable action recognition. 2) Template matching relies on structural and morphological properties of the signals and thus may not apply to the cases where distinguishing patterns are not constituted in morphology of individual sensor streams.

References

[1] H. Ghasemzadeh and R. Jafari, "A greedy buffer allocation algorithm for power-aware communication in body sensor networks," in *Proc. 8th IEEE/ACM/IFIP Int. Conf. Hardware/Software Codesign Syst. Synth.*, New York, NY, USA, 2010, pp.195–204.

[2] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SIGKDD Explor. Newslett.*, vol. 12, no. 2, pp. 74–82, Dec. 2010.

[3] F. Bugan, M. Benedetti, G. Casadio, S. Attala, F. Biagi, M. Manca, and A. Leardini, "Estimation of spatial-temporal gait parameters in level walking based on a single accelerometer: Validation on normal subjects by standard gait analysis," *Comput. Methods Program Biomed.*, vol. 108, no. 1, pp. 129–137, 2012.

[4] L. Palmerini, S. Mellone, G. Avanzolini, F. Valzania, and L. Chiari, "Classification of early-mild subjects with Parkinson's disease by using sensor-based measures of posture, gait, and transitions," in *Artificial Intelligence in Medicine*. New York, NY, USA: Springer-Verlag, 2013, pp. 176–180.

[5] H. Ghasemzadeh and R. Jafari, "Physical movement monitoring using body sensor networks: A phonological approach to construct spatial decision trees," *IEEE Trans. Ind. Informat.*, vol. 7, no. 1, pp. 66–77, Feb. 2011.

[6] S. Patel, K. Lorincz, R. Hughes, N. Huggins, J. Growdon, M. Welsh, and P. Bonato, "Analysis of feature space for monitoring persons with Parkinson's disease with application to a wireless wearable sensor system," in *Proc. 29th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2007, pp. 6290–6293.

[7] C. Lombriser, N. B. Bharatula, D. Roggen, and G. Tröster, "On-body activity recognition in a dynamic sensor network," in *Proc. ICST 2nd Int. Conf. Body Area Netw.*, 2007, pp. 1–6.

[8] E. Guenterberg, H. Ghasemzadeh, V. Loseu, and R. Jafari, "A distributed continuous action recognition using a hidden Markov model on body sensor networks," in *Proc. 5th IEEE Int. Conf. Distrib. Comput. Sens. Syst.*, 2009, pp. 145–158.

[9] E. A. Heinz, K. S. Kunze, M. Gruber, D. Bannach, and P. Lukowicz, "Using wearable sensors for real-time recognition tasks in games of martial arts—An initial experiment," in *Proc. IEEE Symp. Comput. Intell. Games,*2006, pp. 98–102.

[10] E. Sazonov, G. Fulk, J. Hill, Y. Schutz, and R. Browning, "Monitoring of posture allocations and activities by a shoe-based wearable sensor," *IEEE Trans. Biomed. Eng.*, vol. 58, no. 4, pp. 983–990, Apr. 2011.

[11] B. Logan, J. Healey, M. Philipose, E. Tapia, and S. Intille, "A long-term evaluation of sensing modalities for activity recognition," in *Proc. 9th Int. Conf. Ubiquitous Comput.*, 2007, vol. 4717, pp. 483–500.

[12] W. H. Wu, A. A. T. Bui, M. A. Batalin, L. K. Au, J. D. Binney, and W. J. Kaiser, "Medic: Medical embedded device for individualized care," *Artif. Intell. Med.*, vol. 42, no. 2, pp. 137–152, 2008.

[13] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. New York, NY, USA: Wiley-Interscience, 2000.

[14] L. Klingbeil and T. Wark, "A wireless sensor network for real-time indoor localisation and motion monitoring," in *Proc. 7th Int. Conf. Inf. Process. Sens. Netw*, Washington, DC, USA, 2008, pp. 39–50.

[15] L. Bao and S. S. Intille. (2004, Apr.). Activity recognition from user-annotated acceleration data. *Pervasive Comput.*[Online]. pp. 1–17. Available: http://dx.doi.org/10.1007/b96922

[16] J. Lester, T. Choudhury, and G. Borriello. (2006). *A Practical Approach to Recognizing Physical Activities*. Berlin, Germany: Springer-Verlag, [Online]. Available: http://dx.doi.org/10.1007/11748625_1

[17] H. Ghasemzadeh, E. Guenterberg, and R. Jafari, "Energy-efficient information-driven coverage for physical movement monitoring in body sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 1, pp. 58–69, Jan. 2009.

[18] S. Kang, J. Lee, H. Jang, H. Lee, Y. Lee, S. Park, T. Park, and J. Song, "Seemon: Scalable and energy-efficient context monitoring framework for sensor-rich mobile environments," in *Proc. 6th Int. Conf. Mobile Syst., Appl. Serv.*, New York, NY, USA, 2008, pp. 267–280.

[19] P. Zappi, C. Lombriser, T. Stiefmeier, E. Farella, D. Roggen, L. Benini, and G. Troster, "Activity recognition from on-body sensors: Accuracy-power trade-off by dynamic sensor selection," in *Proc. 5th Eur. Conf. Wireless Sens. Netw.*, 2008, vol. 4913, pp.17–33.

[20] P. Veltink, L. Martens, and R. Van Lummel, "Detection of static and dynamic activities using uniaxial accelerometers," *IEEE Trans. Rehabil. Eng.*, vol. 4, no. 4, pp. 375–385, Dec. 1996.

[21] C. Lombriser, O. Amft, P. Zappi, L. Benini, and G. Trster, "Benefits of dynamically reconfigurable activity recognition in distributed sensing environments," in *Activity Recognition in Pervasive Intelligent Environments (Atlantis Ambient and Pervasive Intelligence Series)*, vol. 4, L. Chen, C. D. Nugent, J. Biswas, J. Hoey, and I. Khalil, Eds. Amsterdam, The Netherlands: Atlantis Press, 2011, pp. 265–290.

[22] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman, "Activity recognition from accelerometer data," in *Proc. 20th Nat. Conf. Artif. Intell.*, Cambridge, MA, USA, 2005, vol. 20, no. 3, pp. 1541–1546.

[23] T. Huynh and B. Schiele, "Analyzing features for activity recognition," in *Proc. Joint Conf. Smart Objects Ambient Intell., Innovative Context-Aware Serv., Usages Technol.*, 2005, pp. 159–163.

[24] J. Lester, T. Choudhury, and G. Borriello, "A practical approach to recognizing physical activities," in *Proc. 4th Int. Conf. Pervasive Comput.*, 2006, pp. 1–16.

[25] P. Panuccio, H. Ghasemzadeh, G. Fortino, and R. Jafari, "Power-aware action recognition with optimal sensor selection: An adaboost driven distributed template matching approach," in *Proc. 1st ACM Workshop Mobile Syst., Appl., Serv. Healthcare*, New York, NY, USA, 2011, pp. 5:1–5:6.

[26] J. Lewis, "Fast normalized cross-correlation," *Vis. Interface*, vol. 10, no. 1, pp. 120–123, 1995.

[27] G. Fortino, R. Giannantonio, R. Gravina, P. Kuryloski, and R. Jafari, "Enabling effective programming and flexible management of efficient body sensor network applications," *IEEE Trans. Human-Mach. Syst.*, vol. 43, no. 1, pp. 115–133, Jan. 2013.

[28] N. Raveendranathan, S. Galzarano, V. Loseu, R. Gravina, R. Giannantonio, M. Sgroi, R. Jafari, and G. Fortino, "From modeling to implementation of virtual sensors in body sensor networks," *IEEE Sens. J.*, vol. 12, no. 3, pp. 583–593, Mar. 2012.

[29] R. Polikar, "Ensemble based systems in decision making," *IEEE Circuits Syst. Mag.*, vol. 6, no. 3, pp. 21–45, Jul.–Sep. 2006.

[30] K. Murty and S. Kabadi, "Some NP-complete problems in quadratic and nonlinear programming," *Math. Program.*, vol. 39, no. 2, pp. 117–129, 1987.

[31] H. Ghasemzadeh and R. Jafari, "Data aggregation in body sensor networks: A power optimization technique for collaborative signal processing," in *Proc. 7th Annu. IEEE Commun. Soc. Conf. Sens. Mesh Ad Hoc Commun. Netw.,* Jun. 2005, pp. 1–9.