Opportunistic Hierarchical Classification for Power Optimization in Wearable Movement Monitoring Systems

Francesco Fraternali[1], Mahsan Rofouei[2], Nabil Alshurafa[2], Hassan Ghasemzadeh[2], Luca Benini[1], Majid Sarrafzadeh[2]

[1] Department of Electronic Informatic and System, University of Bologna, Italy
francesco.fraternali@studio.unibo.it, luca.benini@unibo.it

[2] Department of Computer Science, University of California, Los Angeles, CA, USA
{mahsan, nabil, hassan, majid}@cs.ucla.edu

***Abstract*— Patient monitoring systems are becoming increasingly important in accurately diagnosing and treating growing worldwide chronic conditions especially the obesity epidemic. The ubiquitous nature of wearable sensors, such as the readily available embedded accelerometers in smart phones, provides physicians with an opportunity to remotely monitor their patient's daily activity. There have been several developments in the area of activity recognition using wearable sensors. However, due to power constraints, resource efficient algorithms are necessary in order to perform accurate real-time activity recognition while consuming minimal energy. In this paper, we present a two-tier architecture for optimizing power consumption in such systems. While the first tier relies on a hierarchical classification approach, the second one manages the activation and deactivation of the classification system. We demonstrate this using a series of binary Support Vector Machine classifiers. The proposed approach, however, is classifier independent. Experimenting with subjects performing different daily activities such as walking, going up-stairs and down-stairs, standing and sitting, our approach achieves a power savings of 87%, while maintaining 92% classification accuracy.**

Keywords- Mobile Phone; Accelerometer; Support Vector Machines; Hierarchical Classifier; Activity Monitoring; Power Optimization;

I. INTRODUCTION

Remote monitoring systems that use embedded sensors to monitor patients' activities permit doctors to passively gain useful knowledge about the habits of millions of users. These monitoring systems can be constructed using readily available wearable systems that provide continuous acquisition of sensor data. Many prior studies on wearable body sensor networks show promising results in detecting physical activity using accelerometers positioned on one or more locations of the human body [6, 7]. Furthermore, with the availability of tri-axis accelerometers in mobile phones, monitoring is made possible in a less stressful and pervasive manner.

Since wearable sensors and especially mobile phones run on limited supply of power, monitoring systems need to be energy-efficient. For this reason developing embedded signal processing and activity recognition algorithms must be optimized to maximize the effective lifetime of such systems.

There are existing devices that attempt to calculate caloric expenditure, however they require the subject to carry added hardware and do not take into account the subjects' activity type when calculating energy expenditure. Being able to classify and distinguish between activities that provide high vs. low energy expenditure is critical when accurately computing caloric expenditure. In this paper, we attempt to detect and classify a subset of human activities and postures performed on a regular basis that require varying sampling rates. Activities such as walking, going upstairs or downstairs would potentially require higher degrees of signal resolution and sampling rates compared to stationary activities, such as sitting or standing. Furthermore, human subjects tend to perform each activity over an extended period of time, which allows for the deactivating of signal processing algorithms while maintaining acceptable classification accuracy.

The main reasons one does not need to perform continuous data processing for a subject throughout a typical day is: 1) the rate of activity change of a subject in a given day is not high. In particular, in goal-oriented clinical studies, patients carry their activity monitoring only during the period during which the specific actions are performed; 2) misclassification for a small time period (e.g., few seconds) is acceptable and negligible in many applications such as calculating a subjects total energy expenditure for an entire day. For this reason the classifier can execute in a sparse manner when: 1) we want to monitor activity for a long period of time; 2) the frequency of transitioning between activities is small. Based on the above motivation we introduce a novel power saving methodology that optimizes computation and sensing power while maintaining a lower bound on the accuracy of the system. Because the number of samples required to classify each activity varies from one activity to another, one can dynamically adjust the length of the window during which the classification algorithm is active in order to achieve longer battery life, without decreasing the accuracy of the activity recognition system.

978-1-4673-2684-1/12/$31.00 ©2012 IEEE

While our claim for power reduction in this paper is independent of the classifier, we use a supervised learning classifier which implements a support vector machine (SVM) model in performing activity recognition. The SVM model has overwhelming support in the literature for classifying physical activity through the use of an accelerometer sensor [9, and 11]. In comparative studies of motion recognition, SVM classifiers have achieved accuracies of 94.8% and 98.15% [9].

We show that the signal processing architecture can be improved for power savings by adding two layers of computation reduction in the system. Therefore, our proposed computing model is a two-tier signal processing architecture. In the first tier, we introduce a hierarchical classifier that minimizes the amount of computation by reducing the number of simple binary classifiers that run in real-time in order to classify human movements. This hierarchical model is inspired by the fact that human activities occur with different probabilities. For example, activities such as walking are more likely to occur than kneeling. Thus, classifiers that detect activities of higher probabilities would be placed in early stages of the hierarchy. The second tier in our architecture minimizes the power consumption of the sensing system by regular activation and deactivation of the classifier.

The paper is organized as follows: In Section II we provide background information on the process and models used in classification and activity recognition. In Section III we provide a survey of related works that relate to activity monitoring. In Section IV we discuss our main contribution in reducing power consumption in the area of activity monitoring. In Section V we present the experiment and results in proving our proposed approach to power saving in activity recognition, followed by conclusions and future work in Section VI.

II. BACKGROUND

Prior to presenting details of our optimization approach, we provide an introduction of the sensing platform and pre-processing used in capturing signals in Section II.A, a description of the hierarchical approach to classification in Section II.B and an overview of Support Vector Machine (SVM) classifiers, chosen to prove the potential of our proposed algorithm, in Section II.C.

A. Activity Recognition Process

Wearable sensors consist of body-worn sensor nodes that vary by bodily parameters and actions. A typical sensor node used in activity recognition is the motion sensor with embedded gyroscopes and accelerometers. An accelerometer is a device that measures proper acceleration, which is the acceleration associated with the phenomenon of weight experienced by any test mass at rest in the frame of reference of the accelerometer device. It provides a measurement value for each of the three axes in relation to its frame of reference. In this paper a mobile phone is used for data collection and validation of our power optimization approach. Accelerometers are typically embedded in smart phones, which is the chosen platform in proving the feasibility of our power-optimization approach.

The process of classification is illustrated in Fig. 1. Continuous signals from the sensor nodes go through a chain of embedded signal processing blocks, each of which attempt to reduce noise, capture the necessary information from the signal, and extract important features in order to help classify each specific activity.

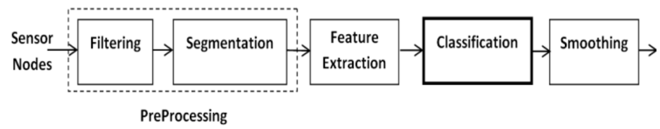


Fig 1. Typical signal processing flow used for classification applications

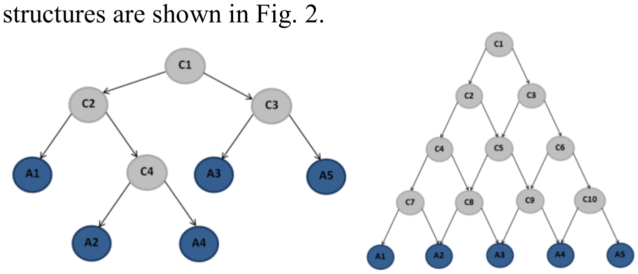
The system works in the following way: the signal sampled by each sensor node is initially filtered in order to reduce high frequency noise. The segmentation block partitions the signal into segments of interest permitting the system to identify the start and end time of each action being classified. After the pre-processing phase, the signals go through a feature extraction block, which computes multiple features for each segment to distinguish one activity from another. In particular each feature captures a different attribute of the signal such as *peak-to-peak amplitude*, *standard deviation*, and *mean value*. Features are calculated from each sensor forming a feature vector that is used for classification. An essential component that affects the performance of a classifier is the features used, for this reason an appropriate feature extraction algorithm is necessary. The importance of this is confirmed in [4], where the results vary based on three different feature extraction methods.

Finally, a classification algorithm is used in order to classify the different actions performed by the user. One example of a classification structure used in this study is the hierarchical classifier.

B. Hierarchical Classification Approach

A Hierarchical Classifier is a particular structure used in order to classify multi-class problems through decomposition into several binary class problems. Usually, the hierarchical classification approach is based on the use of multiple classifiers that are positioned at different levels in a tree. In particular the first classifier is positioned at the top level, the root, and it distinguishes between two types of activities. The lower level classifiers continue to divide the search space until a leaf node in the tree identifies an activity. This particular technique is adopted in [5], where multiple classifiers are used at different levels of a tree structure in order to accurately identify six different activities.

There are many tree structures that can be used in order to build a hierarchical classifier. Several examples and a comparison of such techniques are presented in [17]. One example for a tree configuration classifier is the Directed Acyclic Graph (DAG) where each node can have more than one parent, while another comparable structure is the Binary Hierarchical Classifier (BHC). Examples of these two structures are shown in Fig. 2.



(a) (b)

Fig 2. Examples of Hierarchical Classifiers. Internal nodes represent classifiers (identified by the letter C), and leaf nodes represent actions (identified by the letter A). (a) A typical BHC tree (b) A typical DAG tree.

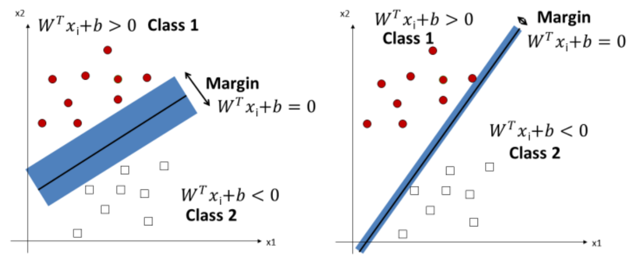
In Fig. 2, internal nodes of the tree represent classifiers, while leaf nodes represent actions. Consequently, if we want to identify *N* activities, we need an *N*-class classifier that has *N* leafs; however the number of classifiers that an activity will have to be tested against depends on the tree structure used. Fig. 2.a shows that a BHC tree needs *(N-1)* binary classifiers for classification of *N* activities, while Fig. 2.b shows that ten classifiers, or $N * (N-1)/2$, are necessary in a DAG tree structure to classify *N* activities. Decreasing the amount of classifiers an activity goes through will greatly increase the speed of the classifier, ultimately effecting the power consumption of an activity recognition system.

There are several types of classification models that can be used in place of the internal nodes in a tree structure. However, our power-saving approach is independent of the classifier model, and for the purposes of proving our concept we use the Support Vector Machine (SVM) model.

C. Support Vector Machines

Support Vector Machines (SVMs) [11,4,5] are statistical learning machines that build an approximated map between samples drawn from an input space and a set of labels (classification) or real values (regression). SVM works by projecting the samples in a multidimensional space and aims to find a separating hyper-plane in this space. A simple two-class example of this kind of separation is shown in Fig. 3.

In Fig. 3, each sample point is represented by x_i, y_i, where x_i in the two-dimensional case represents $[x_{1_i}, x_{2_i}]$ and y_i represents the class the sample point belongs to, $y = 1$ for the class 1, and $y = -1$ for class 2. The objective of SVM is to find the values w and b for a hyper-plane, such that for a set of points **x**, $w^T\boldsymbol{x} + b = 0$, essentially separating the data by maximizing the margin between the classes. The greater the margin between the two classes the better the SVM classifier is able to discriminate between the data.



(a) (b)

Fig 3. SVM Classification: Red dots represent one type of activity and white dots represent another. (a) Shows a greater margin (b) Smaller margin

The process that finds the best hyper-plane is iterative and depends on the number and type of inputs **x**. In particular, the greater the number of inputs used to calculate the hyper-plane the more accurate it will be..

Once the optimal hyper-plane is found, for any unknown data point in the feature space (e.g. **x**), the value of $w^T\boldsymbol{x} + b$ determines to which class (y) the unknown point belongs. Finding the optimal hyper-plane that maximizes the SVM margin involves solving a quadratic optimization problem, where the solution can be expressed as a linear combination of the training samples.

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i^T \tag{1}$$

We are essentially finding a value α_i associated with each training sample x_i. The value of α_i is non-zero only for the training instances that lie within the boundary of the two classes. That is, all other data points that are further away from the boundary will have $\alpha_i=0$. Furthermore, each non-zero α_i indicates that corresponding x_i is a support vector. The classification is then performed by feeding a given unknown sample x to the following equation and using a simple sign function to determine the class label for x.

$$f(x) = \sum_{i=1}^{n} \alpha_i y_i x_i^T \cdot x + b \tag{2}$$

The SVM classification model is widely used in motion classification due to its ability to accurately distinguish and classify different types of motions and activities.

III. Related Works

Inferring activity from wearable sensors has shown promise in recent research. The work in [6] shows the ability of an SVM classifier to accurately recognize six common postures and activities. The results show an average accuracy of up to 98%. In [7] a physical activity (PA) recognition algorithm for wearable wireless sensor networks is proposed using both ambulatory electrocardiogram and accelerometer signals. The overall recognition performance in these papers ranges from 79.3%

to 97.3% for the classification of nine categories of PA using SVM.

Having subjects wear many sensors in multiple body locations for long periods of time can be quite cumbersome for the user. With the recent ubiquitous power of smart phones, researchers have been trying to infer information purely from the sensors available in them, such as the tri-axis accelerometer embedded in many smart phones. In [1], using a common mobile phone tri-axis accelerometer, six activities were classified with an accuracy over 90%. Another example in [8], the authors present a system for localization and activity recognition, based on a smart phone and a single off-the-shelf wireless accelerometer attached to the waist, yielding classification results above 90% accuracy.

Activity recognition using mobile phones has shown great potential in mobile healthcare [2, 10, and 12]. In [2], the physician is able to monitor the patient to see if the subject is in strict compliance with the doctor's exercise plan, recognizing a range of five daily activities with an accuracy of 82%±6.24% using only the tri-axial accelerometer in a smart phone.

Due to the mobile phone platform, and the overall nature of wireless technology, power consumption is becoming a major concern in developing systems and applications. In [15] systems called PowerBooter and PowerTutor were designed with the goal of enabling power modeling and analysis for more smartphone variants and their users. In [16] methods and tools were developed for collecting and analyzing logs on the smart phone of real activity patterns to characterize the power consumption exhibited by different end-users.

In [3] the benefits of dynamic sensor selection are analyzed in order to use available energy efficiently while achieving desired activity recognition accuracy. In [14] the authors introduce a technique to increase the battery lifetime of a PDA-based phone by reducing its *idle power*, the power a device consumes in a "standby" state. Our research takes this further by developing an approach for reducing the power consumption of an activity recognition system using a single wearable sensor node, while maintaining desired activity recognition accuracy.

IV. POWER SAVING OPPORTUNISTIC CLASSIFICATION

In this Section we present and describe our approach in significantly decreasing the power consumption in an activity recognition system. We show that the signal processing architecture can be improved for power savings by adding two layers that reduce the amount of required classifier computation in the system. In Section IV.A we discuss the first layer of power saving. In Section IV.B and IV.C we discuss our main power saving contribution using activation and deactivation of the classifier.

A. *Power saving using a Hierarchical Classifier structure*

The run time or the amount of instructions that the classifier executes directly impacts the amount of power saved. For this reason it's important, when choosing a particular hierarchical classifier configuration, that one minimize the number of internal classifier nodes. Because human activities occur with different probabilities, one can place the classifiers in relation to the probability of a particular activity happening.

In this way, we can choose the action with highest probability and place it in the higher levels of the tree, while the actions with lower probability end up in the lower levels. For example we can easily assume that given a set of daily activities in a patient's life, the action of walking has a higher probability than going upstairs or downstairs.

The configuration that yielded the best power saving results for us, was the *one-against-all (OAA)* [17,18] hierarchical structure illustrated in Fig. 4. Starting from the root node, the classifier C1 compares the first action with all other activities and either outputs A1 or moves down to another level of classification. The final classifier will decide between two activities, so that all activities are classified.

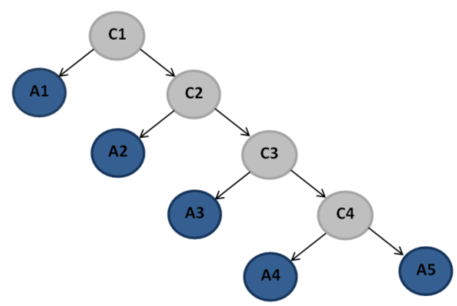


Fig 4. One-against-all (OAA) hierarchical classifier.

Placing activities that occur with higher probability in the higher levels of the tree decreases the amount of executed classifiers. This configuration also simplifies each classifier by decreasing the possible outcomes or class labels. Aside from minimizing instructions, another major effect in power saving is reducing the activation period of the sensing system. As we will demonstrate later in this paper, the hierarchical classification improves the overall accuracy of the system.

B. *Power-saving Opportunistic Classification Heuristic (POCH)*

The second tier of power saving involves the activation and deactivation of the sensor node. Due to the fact that the rate at which a subject transitions from one activity to another is low, we can save power by activating/deactivating the sensor node. Our model analyzes the effects of reducing the activation period on all the activities, and determines an optimal activation/deactivation time, based on a desired trade-off between accuracy and power saving. Our model takes into account two parameters: (1) *T*, which signifies the entire period of activation and deactivation of the sensor and (2) t_d, which signifies only the period of activation of the sensor. An illustration of these parameters is provided in Fig. 5.

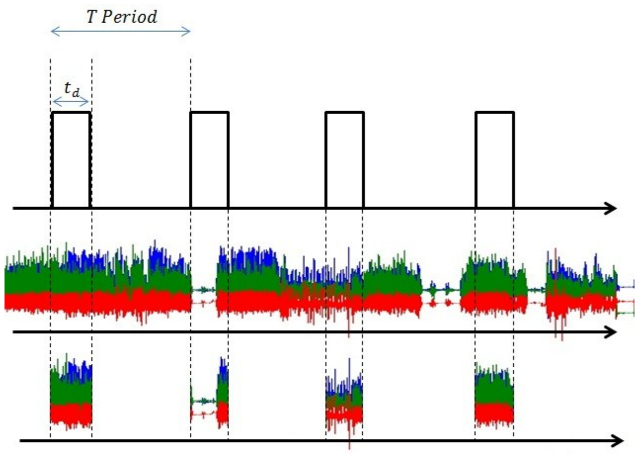


Fig 5. Difference between the parameters T and t_d.

Fig. 5 shows an example of what would happen to the resulting signal if we were to deactivate the accelerometer sensor for a period T-t_d. As shown in the bottom image of Fig. 5, the system will only capture the portion of the activity performed during the activation period t_d. The activity performed during the deactivation period will not be captured. However, if we assume that a subject will transition to another activity during the deactivation period, then the system will capture the new activity during the following activation period t_d, yielding a maximum misclassification error of T-t_d every time the subject transitions from one activity to another. However, when taking a subjects entire day into account, given that T-t_d is much smaller than the entire activity monitoring duration, the misclassification error is negligible.

The value t_d as well as the ratio between T and t_d greatly impact the ability of the classifier to accurately detect certain activities, and could decrease misclassification. In our first experiment, we choose T and t_d that yield a good enough accuracy while maximizing the amount of power saved. T and t_d can be viewed as the duty cycle of the system, where the greater the duty cycle, the greater the overall accuracy of the system, and the greater the power consumption.

C. Adaptive Power-saving Opportunistic Classification Heuristic (A-POCH)

While choosing an overall optimal set of activation parameters yields good results, choosing a set of parameters that optimizes for each activity yields better overall results. For this reason, we decided to dynamically adjust the duty cycle of the system based on the current detected activity. In-fact, we empirically found that each activity type performed will have a classifier optimal value for the deactivation period T-t_d that yields an acceptable misclassification error.

For this reason, we built a state machine that allows the system to dynamically change the value of the deactivation period based on the detected actions. The system starts in an initial state defined by the POCH algorithm in the previous section, where the deactivation period is optimized, but fixed. In this state the system can detect several actions, such as walking, standing, and sitting, going upstairs or downstairs. When a specific action is detected, the system goes into an action-specific state where an action-specific optimal deactivation period is set. Once a new action is detected (the subject is transitioning to a new activity type), the system transitions back to the initial state, and then transitions to its corresponding action-specific state. There is a total of $i+1$ states, where i is the number of activities the system is capable of classifying. Fig. 6 provides an illustration of the state diagram.

Algorithm 1 defines the main power saving method to dynamically activate and deactivate the sensor node using optimal activation parameters. This enables the system to utilize context awareness to feedback a new activation parameter to the sensor node, essentially decreasing the amount of signal processing required and the amount of instructions executed by the classifier.

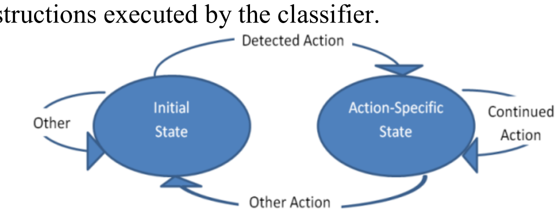


Fig. 6 – A-POCH-state diagram to dynamically adjust the activation / deactivation period of the system.

The Algorithm for this approach is defined below.

Algorithm 1: A-POCH

Input: Activities $A=\{A_1, A_2, ..., A_n\}$;
Initial state: T, t_d
Action-specific state: For each activity i: T_i , t_{di}

Begin:
1. Samples of activity are continuously collected using the initial state activation parameters: T, t_d
2. The system remains in the Initial state until a known activity is detected.
3. Once an activity A_i is detected, the system transitions to an action-specific state and resets sensor node activation parameters to T_i , t_{di}.
4. If the system continues detecting the same action, it remains in the same state. Otherwise it transitions back to the initial state.
5. Repeat Step 1.

V. EXPERIMENTAL RESULTS

A. Experimental Setup

In our experiment we trained our system using six healthy subjects (four men and two women) directed to perform a set of five different activities. We then collected testing data by telling a subject to choose which activities to perform from a set of activities at-will for 30 minutes. These activities are ones that occur frequently in any given day. The activities are listed in Table I.

TABLE I. ACTIVITY DESCRIPTION

Activity ID	*Activity Type*
A1	Walking
A2	Sitting on a chair
A3	Standing
A4	Going Upstairs
A5	Going Downstairs

We asked each subject to place the mobile phone in their pocket while performing the activities. We used a Motorola A956 smart phone that is equipped with the Texas Instruments OMAP3630 - 720 MHz ARM Cortex A8 processor. Due to the nature of smartphones and their scheduling of tasks, an application cannot enforce a strict time requirement on the samples received in the smartphone. However, the accelerometer generated samples at 77 Hz. We logged the data using a simple Android application that stored the tri-axis accelerometers measured values. Once we gathered all the training samples from the different subjects, we processed the data using MATLAB in a common laptop platform. We also used the SVM libraries available in MATLAB to aid in the classification process. In order to obtain the best possible accuracy of our SVM classifier, we had to ensure that the feature extraction mechanism is effective.

B. Classifier Data Analysis

1) Feature Extraction – Pre-Classification

Our objective is to accurately identify each activity performed by the subjects with minimal misclassification errors. We extract signals from the embedded sensor and use a sliding window to extract features. The size of the sliding window indicates the number of measurements used in order to compute the features for training and testing the classifier. The accuracy of the system is greatly affected by the size of the sliding window.

The features collected from the measured signals are *peak-to-peak amplitude*, *standard deviation*, and *mean value*. These features are extracted from each x, y and z-axis of the tri-axis accelerometer. All the features are combined from a given window, and we create a feature vector that is then processed by the classifier. Based on careful analysis of the data we found that the best dimension of the sliding window is 100 measurements, as shown in Fig. 7, sliding at a rate of 10 measurements. Fig. 7 illustrates the accuracy results of our data based on a varying window size.

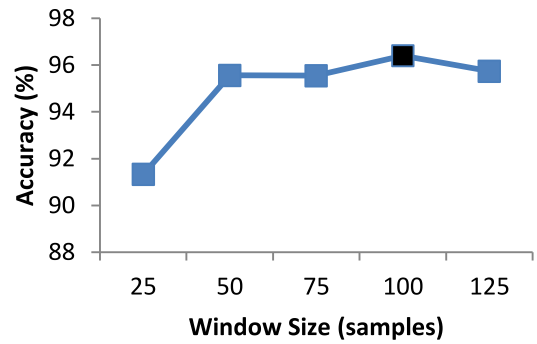


Fig. 7 – Accuracy obtained using different window sizes.

2) Smoothing

During the classification of the movements, a real time software smoothing filter was designed to smooth the results, such that impossible sequence of events are corrected from the output of the classifier. For example, if an individual is walking for 20 seconds, and the classifier misclassifies the 11th second as a run, then it is most probably an error and needs to be corrected. Fig. 8 shows the classifier results before and after smoothing.

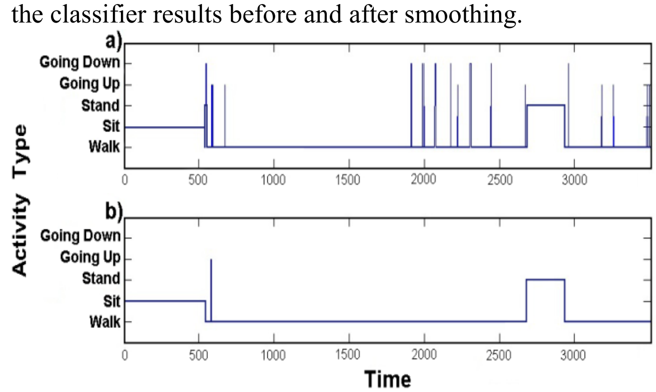


Fig. 8 – Smoothing filter results. (a) Shows the raw results from the SVM classifier before smoothing, and (b) shows the results after smoothing.

This helps decrease the number of misclassifications resulting from the classifier. By smoothing, we are able to increase the accuracy of the activity recognition system from 94.8% to 96.4%.

3) Activation/Deactivation Analysis

Our main contributing factor to power consumption involves the dynamic activation/deactivation of the sensor node and classification algorithm. In order to perform the analysis of our results, we analyzed the effects of deactivating the sensor for a certain period of time, and analyzed the errors that would result from taking such action. When the sensor deactivates, the system maintains the result of the last sample before deactivation. As illustrated in Fig. 9, the top figure shows the deactivation period in red for a particular subject, the middle figure shows the errors resulting from deactivating the sensor node. The bottom figure shows truth. From analysis it can be seen that the misclassification error for a given transition is at-most the size of the deactivation period $T\text{-}t_d$.

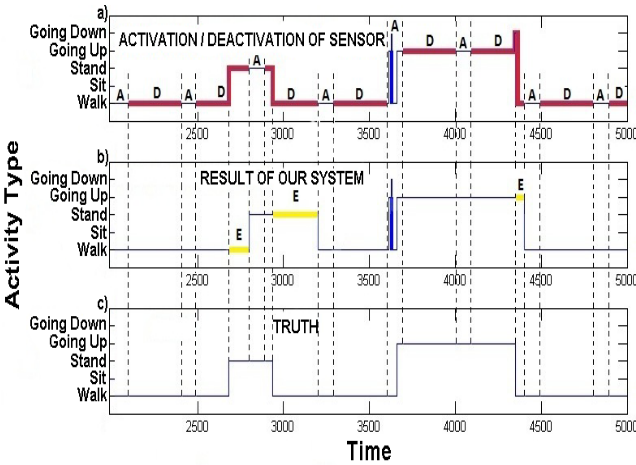


Fig. 9 – Activation/Deactivation analysis. (a) Shows the timing of activating and deactivating a sensor. The letter A indicates an active period, the letter D indicates the sensor is deactivated (in red) (b) shows the errors (in yellow) resulting from activating/deactivating the sensor when compared to (c) where it represents Truth.

C. Hierarchical Classifier Results

Our goal in using the hierarchical classifier is to find the best tree structure that would minimize power consumption. We show how the BHC tree structure can be used in minimizing the expected number of instructions executed during run-time, while meeting system-timing constraints.

We assume a set of *N* possible activities A= {A_1, A_2,..., A_N} each following a probability distribution of P = { p_1, p_2,..., p_N}, based on the probability of a particular activity taking place in a day. We also define l_i which represents the number of instructions executed for the *i*-th activity. The farther the activity node is from the root of the tree, the more classifiers it has to go through and the larger l_i becomes. We introduce in (3) a parameter *E(I),* which is the expected number of instruction executed during run-time.

$$E(I) = \sum_{i=1}^{n} p_i . l_i \tag{3}$$

In order to make a comparison between different tree structures we assume that each classifier in the tree executes the same number of instructions. Using data from the 2010 American Time Use Survey, the probability distributions of six leisure activities for elderly patients are [19]: p_1=62.2%, p_2=12.6%, p_3=9.8%, p_4=7%, p_5=5.6%, p_6=2.8% (respectively for each corresponding activity A_1, A_2, A_3, A_4, A_5, A_6), Table II illustrates the advantage of using the one-against-all (OAA) classifier in comparison to the other classifiers.

TABLE II. EXPECTED VALUE OF INSTRUCTION

Hierarchical Classifier	*E(I)*
Generic BHC	225
DAG	500
OAA	189

Table II. Expected number of instructions executed using different tree structures.

The ARM Cortex A8 processor used by the mobile phone is founded on the ARMv7 architecture and has the ability to scale in speed from 600MHz to more than 1GHz. The Cortex-A8 processor can meet the requirements for power-optimized mobile devices needing operations less than 300mW; and performance-optimized consumer applications requiring 2000 Dhrystone MIPS. Using this along with *E(I)* we can find the energy expended for a single instruction and ultimately for a particular tree structure. Fig. 10 illustrates the power saved using the OAA hierarchical classifier.

Using the OAA hierarchical classifier, we save about 16% in power usage when compared with a typical BHC, and we save up to 62.2% when compared to the DAG tree.

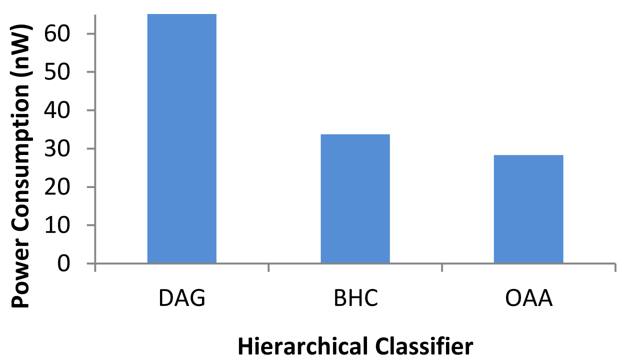


Fig. 10 – Power consumption values using hierarchical classifiers.

Using a hierarchical classifier as opposed to a single classifier enhances the accuracy of the activity recognition system substantially. Fig. 11 shows the accuracy of the classification using a single SVM classifier, without using any hierarchical approach compared to the use of a hierarchical classifier that instead greatly enhances the accuracy of the activity recognition system. In particular Fig. 11 shows that the use of a OAA hierarchical classifier greatly enhances the accuracy of all the subjects, and especially subjects 1 and 2.

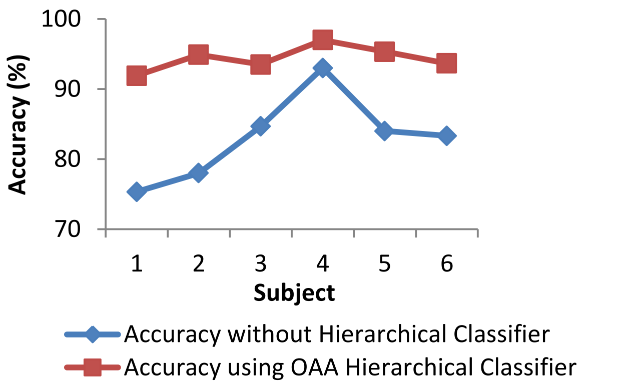


Fig. 11 – Comparison between classification accuracy without using any hierarchical approach, and using OAA binary hierarchical approach .

D. *Power-saving Opportunistic Classification Heuristic (POCH)*

In the POCH algorithm we fix the activation parameters. However, we performed several tests to find the optimal ratio of *T* and t_d that finds the best trade-off between accuracy and power saved. Fig. 12 shows the trade-off between accuracy and power saved. In these tests we considered that one t_d corresponding to the number of samples within the moving time window provides the accuracy shown in Fig. 7.

By analyzing the results of the graph, we can readily see that by increasing the deactivation period we decrease the accuracy of our classifier. However, when t_d equals 15 we can see that the accuracy goes down from 96% (without power saving), to 91% (with power saving). Yet the amount of power saved is about 83.3%. Losing 5% accuracy helps us save 83.3% in power consumption. The power consumption can decrease further if we use T-t_d $=10t_d$ (91% in saving power) but in this case the accuracy of the system goes down to 87%. Depending on the application of the system one can

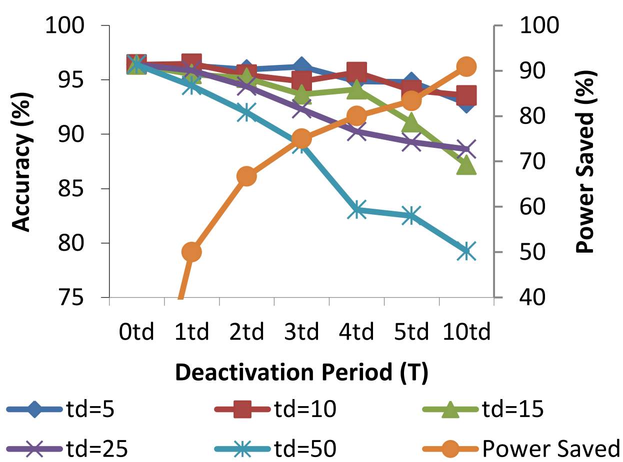


Fig. 12 – Trade-off between accuracy and power saved when varying the duty cycle. The x-axis defines the ratio between the deactivation period (T) and the activation period t_d.

define an acceptable trade-off between accuracy and power consumption and set the activation parameters accordingly.

In order to quantify the amount of power that we can effectively save, we use the values of the accelerometer consumption defined in [13]. The assumption is that the amount of power consumed for a common accelerometer sensor embedded in a mobile phone is an average of $P_1 = 141mW$ per second when the sensor is turned on and an average of $P_2 = 62mW$ per second when the sensor is turned off. Turning on the accelerometer for 5 minutes consumes more energy than activating GPS for 1 minute. Another important number used in quantifying the amount of power saved is the number of samples measured by the accelerometer. Using an average measurement rate of 77Hz, we define the time coverage for each feature based on the window size. The following equation shows how we quantify the amount of energy consumed during each period, where t_{on} is the time the sensor is on and t_{off} is the time the sensor node is off.

$$E_T = t_{on}P_1 + t_{off}P_2 \quad (3)$$

When setting t_d =15, we show the amount of power that can be saved. The effective power saved is about 83% for *T=5td* and 91% for *T=10td*.

E. *Adaptive Power-saving Opportunistic Classification Heuristic (A-POCH)*

In A-POCH, the system dynamically modifies the activation parameters based on the current activity detected. We carefully tested and analyzed our algorithm to determine an optimal *T* and t_d for each activity. Fig. 13 illustrates the power saving and accuracy of the system using two activities walking and standing.

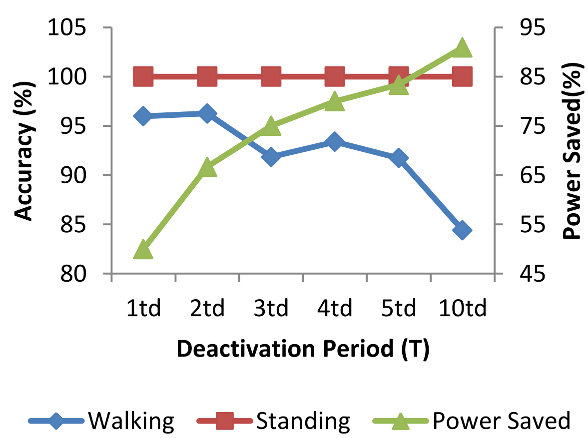


Fig. 13 - Accuracy and power-saving using the A-POCH Algorithm.

As shown in Fig. 13, by increasing the deactivation period from 2td to 5td, the accuracy of the system decreases by 5%, however the added power saved is 16.67% (from 66% to 83.33%). A similar situation for the classification of standing where setting the deactivation to 10td doesn't change the accuracy of the system but provides a power saving of 91%. The more one changes the deactivation period the higher the misclassification error.

Fig. 14 also provides the effective power consumption saved for all the activities, comparing the power saved and accuracy of the POCH and the dynamic A-POCH algorithm. As shown in the Fig. 14, the use of the A-POCH algorithm increases the percentage of power saved from 83% to 87%, while maintaining better classification accuracy. In relation to the POCH algorithm, the A-POCH algorithm increases the deactivation period for the following activities: walking, standing and going upstairs. We also realized that decreasing the deactivation period for the going downstairs activity resulted in higher accuracy. This could be due to the nature of the subject's activities; different subjects will transition from one activity to the next with varying probabilities. This algorithm shows potential for a dynamic context-aware approach that would control the activation period of the embedded sensor node.

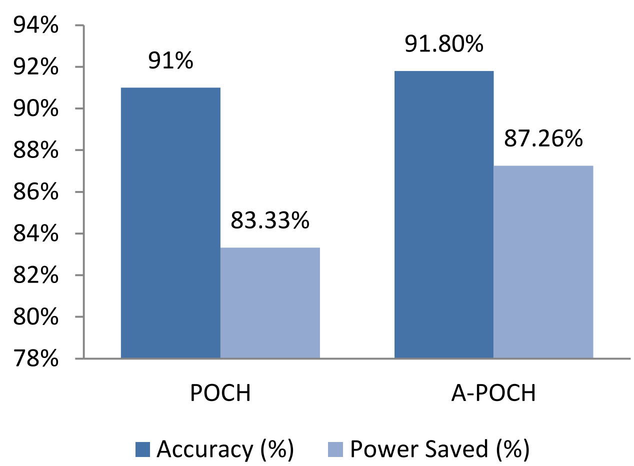


Fig. 14 - Accuracy and power saved comparison using the POCH and A-POCH algorithm and setting t_d =15.

VI. CONCLUSION AND FUTURE WORK

In this work we present and demonstrate two power saving approaches with the goal to seriously decrease the power consumption of the entire activity recognition system. In particular we efficiently implemented a SVM OAA hierarchical classification algorithm to classify five common daily activities using only a tri-axis accelerometer embedded in the mobile phone. We showed in Section V that using a simple OAA hierarchical classifier can add close to 16% power savings.

As shown in Section V, using the second power saving approach we save 83% (using the POCH algorithm) and 87% (using the dynamic A-POCH algorithm) power consumption using a context-aware activation/deactivation of the sensor node, while maintaining an accuracy of 92%.

In future work, we would like to analyze the amount of power saved on a larger set of activities, while conducting a large-scale study. We would further explore power saving options using different types of embedded sensor nodes and implementing the classification locally on the mobile nodes in order to better evaluate the results obtained in this paper. In fact, the classification is currently done off-line in MATLAB. As part of our future work, we will implement the proposed algorithms on the mobile phone and reassess the performance of the system in terms of power consumption. However, our results show the ability of our architecture to minimize the power consumption of the sensing system by regular activation and deactivation of the classifier. Our system shows potential in enabling physicians to monitor the activity of their patients in an energy efficient manner.

VII. REFERENCES

[1] Jennifer R. Kwapisz, Gary M. Weiss, and Samuel A. Moore. 2011. Activity recognition using cell phone accelerometers. SIGKDD Explor. Newsl. 12, 2 (March 2011), 74-82.

[2] Zhongtang ZHAO et. al., "Cross-People Mobile-Phone Based Activity Recognition,". In Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, pages 2545-2549.

[3] Piero Zappi, Clemens Lombriser, Thomas Stiefmeier, Elisabetta Farella, Daniel Roggen, Luca Benini, Gerhard Tröster, "Activity Recognition from On-Body Sensors: Accuracy-Power Trade-Off by Dynamic Sensor Selection Wireless Sensor Networks," In Wireless Sensor Networks, Vol. 4913 (2008), pp. 17-33

[4] Zhenyu He; Lianwen Jin; Lixin Zhen; Jiancheng Huang; , "Gesture recognition based on 3D accelerometer for cell phones interaction," Circuits and Systems, 2008. APCCAS 2008. IEEE Asia Pacific Conference on, vol., no., pp.217-220, Nov. 30 2008-Dec. 3 2008

[5] Shumei Zhang; McCullagh, P.; Nugent, C.; Huiru Zheng; , "Activity Monitoring Using a Smart Phone's Accelerometer with Hierarchical Classification, "Intelligent Environments (IE), 2010 Sixth International Conference on , vol., no., pp.158-163, 19-21 July 2010

[6] Sazonov, E.S.; Fulk, G.; Hill, J.; Schutz, Y.; Browning, R.; , "Monitoring of Posture Allocations and Activities by a Shoe-Based Wearable Sensor," Biomedical Engineering, IEEE Transactions on , vol.58, no.4, pp.983-990, April 2011

[7] Li; Rozgić, V.; Thatte, G.; Sangwon Lee; Emken, A.; Annavaram, M.; Mitra, U.; Spruijt-Metz, D.; Narayanan, S.; , "Multimodal Physical Activity

Recognition by Fusing Temporal and Cepstral Information," Neural Systems and Rehabilitation Engineering, IEEE Transactions on , vol.18, no.4, pp.369-380, Aug. 2010

[8] Martin, E.; Shia, V.; Posu Yan; Kuryloski, P.; Seto, E.; Ekambaram, V.; Bajcsy, R.; , "Enhancing context awareness with activity recognition and radio fingerprinting," *Semantic Computing (ICSC), 2011 Fifth IEEE International Conference on* , vol., no., pp.263-266, 18-21 Sept. 2011

[9] Lin Sun, Daqing Zhang, Bin Li, Bin Guo, and Shijian Li. 2010. Activity recognition on an accelerometer embedded mobile phone with varying positions and orientations. In*Proceedings of the 7th international conference on Ubiquitous intelligence and computing*(UIC'10), Zhiwen Yu, Ramiro Liscano, Guanling Chen, Daqing Zhang, and Xingshe Zhou (Eds.). Springer-Verlag, Berlin, Heidelberg, 548-562.

[10] Bruce H. Dobkin, et al. Reliability and Validity of Bilateral Ankle Accelerometer Algorithms for Activity Recognition and Walking Speed After StrokeStroke. 2011;42:2246-2250

[11] Nishkam Ravi, Nikhil Dandekar, Preetham Mysore, and Michael L. Littman. 2005. Activity recognition from accelerometer data. In *Proceedings of the 17th conference on Innovative applications of artificial intelligence - Volume 3* (IAAI'05), Bruce Porter (Ed.), Vol. 3. AAAI Press 1541-1546.

[12] Toshiki Iso and Kenichi Yamazaki. 2006. Gait analyzer based on a cell phone with a single three-axis accelerometer. In *Proceedings of the 8th conference on Human-computer interaction with mobile devices and services* (MobileHCI '06). ACM, New York, NY, USA, 141-144.

[13] Jeongyeup Paek, Joongheon Kim, and Ramesh Govindan. 2010. Energy-efficient rate-adaptive GPS-based positioning for smartphones. In *Proceedings of the 8th international conference on Mobile systems, applications, and services* (MobiSys '10). ACM, New York, NY, USA, 299-314.

[14] Eugene Shih, Paramvir Bahl, and Michael J. Sinclair. 2002. Wake on wireless: an event driven energy saving strategy for battery operated devices. In *Proceedings of the 8th annual international conference on Mobile computing and networking* (MobiCom '02). ACM, New York, NY, USA, 160-171.

[15] Lide Zhang, Birjodh Tiwana, Zhiyun Qian, Zhaoguang Wang, Robert P. Dick, Zhuoqing Morley Mao, and Lei Yang. 2010. Accurate online power estimation and automatic battery behavior based power model generation for smartphones. In *Proceedings of the eighth IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis* (CODES/ISSS '10). ACM, New York, NY, USA, 105-114.

[16] Alex Shye, Benjamin Scholbrock, and Gokhan Memik. 2009. Into the wild: studying real user activity patterns to guide power optimizations for mobile architectures. In *Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture* (MICRO 42). ACM, New York, NY, USA, 168-178.

[17] Hsu and C. Lin. A comparison of methods for multi-class support vector machines. *IEEE Transactions on Neural Networks*, 13:415{425, 2002.

[18] Ryan Rifkin and Aldebaro Klautau. 2004. In Defense of One-Vs-All Classification. *J. Mach. Learn. Res.* 5 (December 2004), 101-141.

[19] American Time Use Survey. (2010). *American Time Use Survey*. Retrieved from Bureau of Labor Statistics website:http://www.bls.gov/tus/charts/LEISURE.HTM