Movement Recognition Using Body Area Networks

John Paul Varkey and Dario Pompili
Department of Electrical and Computer Engineering
Rutgers University, Piscataway, NJ 08854
Emails: jpvarkey@eden.rutgers.edu, pompili@ece.rutgers.edu

***Abstract*—Significant research has been done on recognizing the daily activities using acceleration data but few works have focused on classifying the movements comprising an activity due to the shorter time scales of the movements compared to that of an activity. Recognizing the individual movements within an activity can help improve the activity recognition on the whole by using the extra information from the movement granularity. Also, for many applications such as rehabilitation, sports medicine, geriatric care, and health/fitness monitoring the importance of movement recognition cannot be overlooked. Hence, in this paper a novel machine learning algorithm using body area networks is proposed that can on the fly, jointly classify the type of movements, and starting and finishing instant of each movement within an activity. A case study on the best set of features and minimum number of accelerometers needed to correctly classify movements within a smoking activity is also presented.**

I. Introduction

Movement recognition systems capable of continuously monitoring and recognizing movements have a variety of applications such as rehabilitation, sports medicine, geriatric care, and health/fitness monitoring. A movement recognition system can help reduce the errors that arise from self-reporting or by paid personals and also enable people to engage in their daily routines in an unimpeded manner. Such systems would also enable remote monitoring of patients through Body Area Networks (BANs), which consist of wearable sensor nodes wirelessly communicating with each other and the Internet.

To understand human motion, it is important to point out the difference between an *activity* and a *movement* that comprises an activity. A physical movement is a body posture/gesture that typically lasts for several milliseconds or seconds, whereas an activity lasts several minutes or hours and comprises of different physical movements that may be repeated over time. For example, a smoking activity would comprise of movements such as the 'arm moving up' for smoking followed by the 'arm moving down' after taking the puff, which are repeated over time until a cigarette gets over. Hence, smoking a cigarette is an activity, whereas moving the arm up and down should be considered as movements. Although there are activities in which it is difficult to identify and separate the movements out because of their shorter time scale and high correlation between the individual movements, the importance of movement recognition for many applications cannot be overlooked. Moreover, recognizing the individual movements within an activity can help improve the activity recognition on the whole by using the extra information from the movement granularity.

There has been considerable research on activity recognition using acceleration data compared to recognizing the individual movements. Previous works on activity recognition using acceleration values have considered features like mean, standard deviation, maximum, peak-to-peak, root-mean-square, and correlation of acceleration values between pair of accelerometer axes [1]–[5]. Conversely, in this paper we focus on movement recognition, which is a more difficult problem to deal with as, in general, shorter time scales are involved. In [5], the authors show that - like any spoken language - movements also have a grammatical framework. Their method consists of assigning certain primitives to each movement and then exploit the notion of a decision tree to identify atomic actions corresponding to every given movement. However, while this approach is very interesting, the algorithm proposed in [5] does not identify the starting and finishing instants of movements within an activity, which are assumed to be known a priori.

The computer vision community has done research on human motion recognition using time frames of a video sequence [6], [7]. However, those techniques require external infrastructure - e.g., (infrared)cameras - and may be biased by environmental conditions such as background light or heat. Also, such techniques cannot be directly applied to those scenarios that require privacy, even if the image is blurred. Analysis in [8], [9] has shown that inertial sensors like accelerometers, which provide acceleration values of body motion [1], [10], outperform most sensors in human motion recognition. Besides the fact that they usually lead to good results in motion recognition, they are small and less expensive than cameras, require relatively little energy, and are fairly insensitive to environmental conditions. Moreover, movement recognition systems using accelerometers can be tuned to recognize only the set of movements involved in a specific activity. This feature can be exploited to guarantee some form of privacy as not all the actual movements would be recognized or recorded by the system, but only some predefined ones.

In this paper, we introduce a novel machine learning algorithm that can on the fly, jointly classify the type of movements, and starting and finishing instant of each movement within an activity. We chose machine learning technique over a deterministic approach to classify movements within an activity as two movements of the same type (even performed by the same person at close time instants) would never be exactly the same. Also, for the specific activity of 'smoking', which we use as a case study, we identify the best set of features from the acceleration values of movements, the number of sub-intervals at which those features are extracted, and the minimum number of accelerometers to classify movements. Although we assess the performance of our algorithm considering a specific case study, the methodology proposed in this work can be applied to a broad range of activities.

The remainder of this paper is organized as follows. In Sect. II, we discuss the different phases to perform movement recognition. In Sect. III, we introduce the novel algorithm to classify the type of movements, while in Sect. IV we present a case study on the best set of features and the minimum number of accelerometers needed for classifying movements within a smoking activity. Finally, in Sect. V, we draw the main conclusions.

II. Coordinated Movement Recognition

In this section, we discuss the various phases involved in coordinately classifying movements in BANs. Our approach involves four phases - (1) Feature Extraction, (2) Training, (3) Classification, and (4) Tuning.

A. Feature Extraction

Initially, a large amount of raw acceleration data is collected for the different movement types to be classified at a sampling frequency of 140 Hz. In addition, many observations of each movement type are collected. However, using all of this information would be inefficient and would add to the complexity of the algorithm. In order to help the classifier better represent each movement, meaningful features need to be extracted from the raw data not just from the entire time interval of each movement but from N sub-intervals of each movement. A *classifier* is a function that maps input data samples to a defined set of object class after having 'seen' a number of training examples. Features considered for movement recognition are listed in Table I. Also, depending on the number of accelerometers used, there is a tradeoff between cost and comfort of the subject, which makes it essential identifying the minimum number of accelerometers needed to recognize movements without affecting the classification accuracy. Hence, for the case study we considered in this paper (smoking), we used sets of 1, 2, and 3 triaxial accelerometers attached to Imote2 [11] sensor motes on the right arm of the subject at positions given in Table I. The validity of this approach, however, is not limited to the smoking activity.

B. Training

Once the features are extracted from the acceleration data for each movement type to be recognized, the classifier needs to 'familiarize' with the set of movement types by training itself using the features that characterize each of the movement type. To this end, the classifier is fed with features from several observations of each movement type to be recognized. The type of the classifier used can affect the performance of the system. There are basically two types of classifiers - *Support Vector Machines (SVMs)* and *Neural Networks (NNs)*. In SVMs, the classification is achieved using a hyperplane that separates the data samples into different classes (Fig. 1). The closest data points to the hyperplane are called Support Vectors (SVs). Conversely, a NN consists of interconnecting nodes to classify the input data into a defined object class leading to three layers: input, output, and a hidden layer (Fig. 1). Each layer receives one or more inputs and sums them to produce an output value, which represents an object class.

TABLE I
Feature extraction

Features from acceleration values	
$Mean_X, Mean_Y, Mean_Z$	Mean acceleration on x, y, z
Max_X, Max_Y, Max_Z	Maximum acceleration on x, y, z
Std_X, Std_Y, Std_Z	Standard deviation of acceleration on x, y, z
PP_X, PP_Y, PP_Z	Peak-to-peak acceleration on x, y, z
RMS_X, RMS_Y, RMS_Z	Root-mean-square acceleration on x, y, z
$Cor_{XY}, Cor_{YZ}, Cor_{XZ}$	Correlation of acceleration for pairs of xy, yz, xz
Triaxial Accelerometer parameters	
Sampling Frequency	140 Hz
Quantity	Sets of 1, 2, and 3 accelerometers
Placement	Wrist, below the elbow, above the elbow

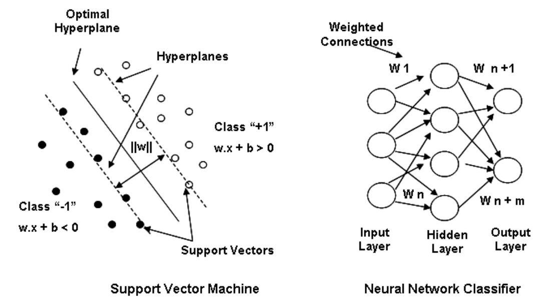


Fig. 1. Type of Classifiers.

The main disadvantage of a NN is that it tends to classify movements accurately only for input data similar to the training samples, whereas in SVMs such 'over-fitting' does not occur as a mathematical kernel is used [12]. Hence, we chose SVMs as classifier for movement recognition. The essence of SVM method is the construction of an optimal hyperplane, which can separate data from opposite classes using the biggest possible margin. Feeding the SVM with several observations of the movement types to be recognized allows to obtain a hyperplane, which can be defined as,

$$F(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b, \tag{1}$$

where $\mathbf{w}$ is the vector normal to the hyperplane, $\mathbf{x}$ is a general vector, and b is a constant that shifts the hyperplane. This function is not suitable for solving more complicated, linearly non-separable problems. That is where kernel functions come into play. Rather than fitting nonlinear curves to the data, SVM handles this by using a kernel function to map the data into a high-dimensional space where a hyperplane can be used to do the separation. There are four basic kernels - linear, polynomial, radial basis function (RBF), and sigmoid, out of which we used the RBF kernel for movement recognition as it showed to be more robust.

C. Classification

The typical architecture of a BAN that coordinately classify movements (Fig. 2(a)) involves three categories of nodes - *basic nodes*, *cluster heads (CH)*, and a *sink node*, S. Each node

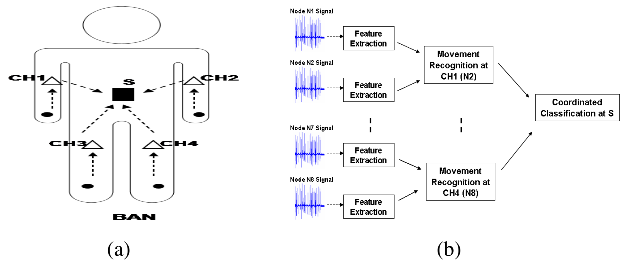


Fig. 2. (a) BAN; (b) Sequence flow of coordinated movement recognition.

(except the sink) can sample signals and extract features from the acceleration values. The basic nodes will then transmit those features wirelessly to the CH, using a Time Division Multiple Access (TDMA) communication protocol, which will then classify the movements locally. For example, recognizing the right arm movement can be done locally at CH2. Each CH can locally recognize the sub-movements and then send that information to the sink node, where the collective recognition is performed. Figure 2(b) shows the sequence flow of coordinated movement recognition.

D. Tuning

Tuning phase falls in between the Training and the Classification phases. In this phase, the system computes the parameters related to the windowing algorithm so as to optimize the support vectors for recognizing the movements with suitable accuracy. We use *cumulative misclassification ratio* as the index for selecting the optimized parameter values, which is the ratio of total misclassification of movements and the total activity time. We consider a recognized movement to be misclassified if i) either the type recognized is not the actual type or ii) the time interval of the recognized interval is less than 20% of the actual movement time interval, which we refer to as *jitter*. This is a conservative definition that enables to accurately assess the performance of the system in real life scenarios. Figure 3 shows the various phases of movement recognition.

III. Problem Formulation

In this section, we explain how each movement types are represented in the training phase and how the windowing algorithm based on acceleration data can on the fly, jointly classify the type of movements, and starting and finishing instant of each movement within an activity. As explained in Sect. II-B, we train the SVM with movement types to be classified using a sample set of activities (i.e., training set), whereas in the classification phase (Sect. II-C) the system recognizes the unknown movements involved in the current activity. Currently, in our approach, the training and the tuning phases are done offline, whereas the classification is done online.

Before we explain the windowing algorithm let us define the parameters used to represent the movements in the training as well as in the classification phase. Let T_A [s] be the total time interval of a set of M movements such that each movement $m = 1, 2, \cdots, M$, and let $\mathcal{A}$ be the training set,

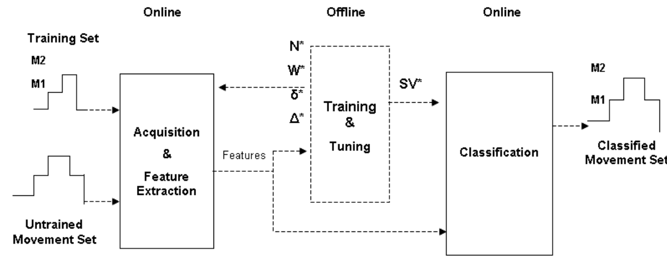


Fig. 3. Movement recognition phases.

i.e., the association of set of movement types with the set of movements. In addition, let $\mathcal{S}$ be the set of sensors in the body area network such that $s = 1, 2, \cdots, |\mathcal{S}|$ and let ζ as given in (2) be the set of R movement types involved such that $\mathcal{T}_r$ be the r^{th} movement type, where $r = 1, 2, \cdots R$. Each movement m can then be associated with any and only one of the movement type in ζ (generally $R \ll M$). Features $\mathcal{F}$ are extracted from N sub-intervals of each movement m such that each feature $f = 1, 2, \cdots, |\mathcal{F}|$. Figure 4(a) shows the physical representation of the m^{th} movement of type $\mathcal{T}_r$ on an axis of the triaxial accelerometer, which is represented by a line as its logical representation in Fig. 4(b). v_m^{sx} in (3) denotes the set of features extracted from the x axis of the accelerometer attached to the generic sensor node s, whereas v_m^s in (4) is the set of features from all the axes of the accelerometer attached to that node. v_m in (4) represents a movement as the set of all features from all the sensors in the body area network in Fig. 2(a). Therefore, the training set $\mathcal{A}$ can be represented as the set of all the movements and its type as in (5).

$$\zeta = \{\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_r, ..., \mathcal{T}_R\}, \tag{2}$$

where $\forall$m $= 1, 2, ..., M$, m $\rightarrow \mathcal{T}_r$ if the m^{th} movement is of type $\mathcal{T}_r^m$.

$$v_m^{sx} = [f_1^1, f_2^1, ..., f_{|\mathcal{F}|}^1; ...; f_1^n, f_2^n, ..., f_{|\mathcal{F}|}^n; f_1^N, f_2^N, ..., f_{|\mathcal{F}|}^N], \tag{3}$$

$$v_m^s = [v_m^{sx}, v_m^{sy}, v_m^{sz}], \ v_m = [v_m^1, v_m^2, ..., v_m^s, ..., v_m^{|\mathcal{S}|}], \tag{4}$$

$$\mathcal{A} = \{(v_1, \mathcal{T}_r^1), ...(v_m, \mathcal{T}_r^m), ..., (v_M, \mathcal{T}_r^M)\}, \ \mathcal{T}_r^m \in \zeta. \tag{5}$$

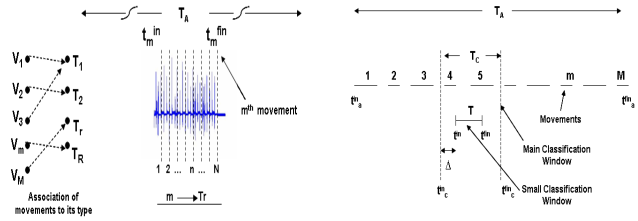
Once the training is done, we derive the normal vector $\mathbf{w}$ to the hyperplane as shown in (6) using the support vectors, and the Lagrange multipliers [13] obtained from the SVM. This vector $\mathbf{w}$ is then used to recognize the movements in the classification phase using the windowing algorithm as,

$$\mathbf{w} = \sum_{i=1}^{N_s} \alpha_i . U_i, \tag{6}$$

where N_s is the number of support vectors, U_i is the i^{th} SV, and α_i is the i^{th} Lagrange multiplier.

Let us define one more term called "confidence". For a vector, say $\mathbf{x}$, its confidence is defined as its distance from the hyperplane ($\mathbf{w}$, b), which is given by

$$d(\mathbf{w}, b; \mathbf{x}) = \frac{\mathbf{w} \cdot \mathbf{x} + b}{||\mathbf{w}||}, \tag{7}$$



(a) (b)

Fig. 4. (a) Physical representation of the m^{th} movement; (b) Classification windows and logical representation of movements.

This distance $d(\mathbf{w}, b; \mathbf{x})$ indicates how confident the classifier is about the class of the vector $\mathbf{x}$. The higher $d(\mathbf{w}, b; \mathbf{x})$, the more certain the classifier is about the class of the vector $\mathbf{x}$. This forms the basic concept of our classification algorithm in recognizing movements using the windowing algorithm.

The algorithm uses two classification windows, a *main* and a *small classification window*. The small classification window moves only within the main classification window. Let the interval T_A and the main classification window T_C [s], respectively, be represented using the starting and finishing time instants as $[t_a^{in}, t_a^{fin}]$ and $[t_c^{in}, t_c^{fin}]$. For the classification phase, we consider two intervals, T^{min} and T^{max}, which are respectively the minimum and the maximum time interval among M movements in the training set,

$$T^{min} = \min_{1 \le m \le M} |t_m^{fin} - t_m^{in}|, \; T^{max} = \max_{1 \le m \le M} |t_m^{fin} - t_m^{in}|, \tag{8}$$

where t_m^{fin} and t_m^{in} are the final and starting time of the m^{th} movement in the training set $\mathcal{A}$, respectively.

The main classification window interval, indicated by T_C in Fig. 4(b), has an interval that is always set to T^{max} and is shifted over time. For $p = 1, 2, \cdots, P$, with $P = \lfloor \frac{T^{max} - T^{min}}{T^{min}/\delta} \rfloor + 1$, there is a small classification window interval, $T[p]$, defined as

$$T[p] = T^{min} + (p-1)\frac{T^{min}}{\delta}. \tag{9}$$

The number of small classification windows within the main classification window, P, depends on T^{min} and δ. The small classification window is shifted by T^{min}/Δ within the main classification window for each $T[p]$ value as given in (9), until it reaches the end of the main classification window. The number of such shifts is represented by K. For each shift of the small classification window $T[p]$ within the main classification window T_C, the confidence is calculated. Once the confidence for all the small classification windows $T[p]$ within the main classification window T_C is computed, we take three small window intervals with best confidence and check for any overlap in their time intervals. If there is any overlapping windows, we combine those time intervals and then compute the confidence for the combined interval. If there is no overlapping windows, then we avoid those window intervals that have confidence less than the average confidence among all the small windows considered within the main classification window. The recognized movement will be the class of the window interval that has best confidence. Once the movement is recognized the main classification window T_C is shifted by the recently recognized time interval. The pseudo code of this windowing algorithm is given in Algorithm 1.

The performance of the algorithm depends on various parameters - (1) Feature set $\mathcal{F}$, (2) Number of sub-intervals N, (3) Number of small classification windows, number of small classification windows P within the main classification window, and (4) Number of shifts of small classification windows K. Therefore, Algorithm 1 requires certain amount of fine tuning using the training set before it can be applied for recognizing unknown movements in an activity as discussed in Sect. II-D. The tuning phase is done off-line just like the training phase. For tuning the system to the right values of $\mathcal{F}$, N, δ, and Δ, we place the main classification window at the exact location of movements in the training set and try to classify the movements by changing values of $\mathcal{F}$, N, δ, and Δ. Once the optimal values for $\mathcal{F}$, N, δ, and Δ as indicated by starred values in Fig. 3 are found, we can obtain the optimal support vectors to classify movements during the classification phase either at the sink node or at a CH.

Algorithm 1 Windowing Algorithm

$P = \lfloor \frac{T^{max} - T^{min}}{\frac{T^{min}}{\delta}} \rfloor + 1$
$t_c^{in} = t_a^{in}$
$t_c^{fin} = t_c^{in} + T^{max}$
while $t_c^{fin} <= t_a^{fin}$ **do**
  for $p = 1$ to P **do**
    $T[p] = T^{min} + (p-1)\frac{T^{min}}{\delta}$
    $K = \lfloor \frac{T^{max} - T}{T^{min}/\Delta} \rfloor + 1$
    for $k = 1$ to K **do**
      $t^{in} = t_c^{in} + (k-1)\frac{T^{min}}{\Delta}$
      $t^{fin} = t_c^{in} + (k-1)\frac{T^{min}}{\Delta} + T$
      Extract features; Find confidence
    end for
  end for
  Take 3 classified small windows with best confidence
  if Any overlapping intervals **then**
    Find confidence of combined intervals
  else
    Avoid intervals with confidence $<$ average
  end if
  Recognized Movement = Class of window that has highest confidence
  $t_c^{in} = t^{fin}$
  $t_c^{fin} = t_c^{in} + T^{max}$
end while

IV. Performance Evaluation

In this section, we present the performance of the algorithm to recognize movements whiting a particular type of activity - smoking - using only acceleration data and also a case study on identifying the minimum number of accelerometers and the best feature set for classifying movements, and other parameters that impact the classification performance of the algorithm. We represent a movement logically by a line with length equal to the time interval for which the movement occurs and each movement type is put at different heights within the activity time interval to differentiate them. We used two indices to evaluate the performance of the algorithm using acceleration data - (1) *Cumulative Misclassification Ratio*, which is the ratio of the total misclassification of movements

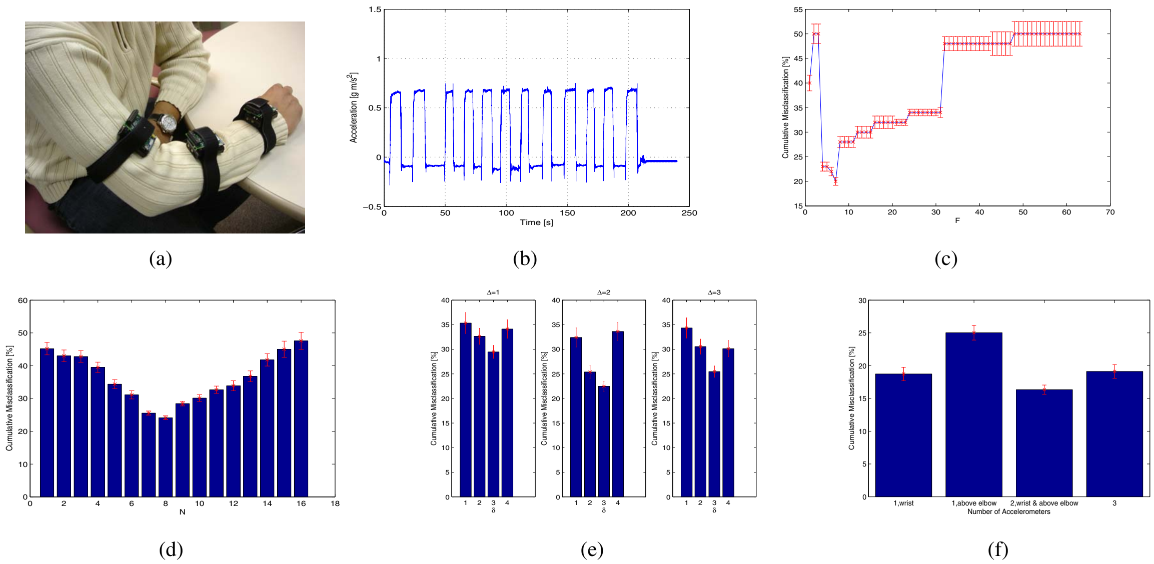


Fig. 5. (a) Three accelerometers on the right arm; (b) x axis acceleration of wrist accelerometer; (c) Cumulative misclassification vs F; (d) Cumulative misclassification vs N; (e) Cumulative misclassification vs δ for different Δ; (f) Cumulative misclassification vs number of accelerometers.

(measured in time) at the current time and the total time, and (2) *Moving Average Misclassification Ratio*, which is the ratio of misclassification of movements over a moving time window. For finding moving average misclassification ratio, we took a time window of interval 10% of the activity time and shifted it over the activity to calculate the misclassification over each window. We used the SVM based machine learning toolbox in Matlab called 'Spider' [14] for movement recognition. Both the training set as well as the set of unrecognized activities were taken from the same subject.

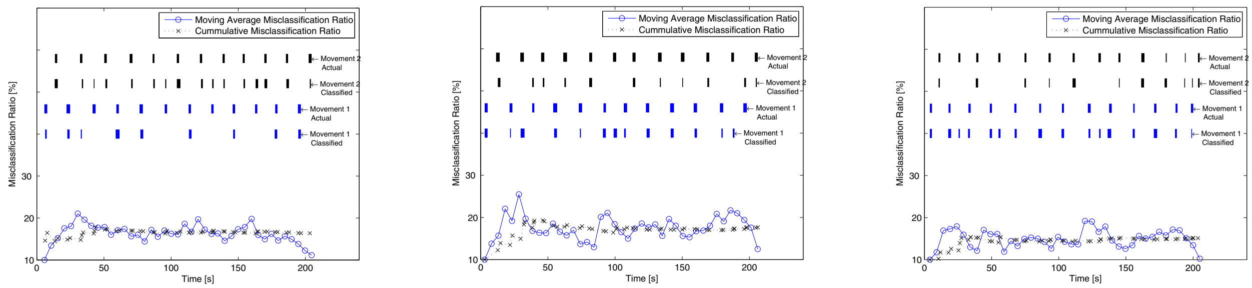
To show the applicability of our algorithm to a real case scenario, we applied the algorithm on a specific activity, smoking. Smoking is health hazard and so studying the smoking pattern of people helps in knowing how adversely smoking affect the health of a person over a period of time. We trained the SVM with a set of 30 observations of smoking activity and classified another set of smoking activities. We consider around $3-4$ minutes as the time to smoke one cigarette by a person and we extracted features such as mean, maximum, standard deviation, peak-to-peak, root mean square of the acceleration values on each of the three axes as well as the correlation of acceleration values between pair of axes from each sub-intervals of the movements. We used sets of 1, 2, 3 accelerometers to help identify the minimum number of accelerometers needed to recognize movements without impairing the accuracy. Figure 5(a) shows three accelerometers attached to the right arm of the subject as well as acceleration values on the x axis of the triaxial accelerometer.

A. Best Set of Features

We considered in total six different features for classifying movements in the smoking activity. In general, the best set of features depends on the type of activity. To find the right set of features, we use 6 binary values in $F = [xxxxxx]_2$ and so we have total of 63 combinations for F. In $\mathcal{F}$, features are considered in the order - mean, maximum, standard deviation, peak-peak, RMS, and then correlation. If a binary element in F is 0, it means that particular feature is excluded, otherwise it is included. So, if $F = [000111]_2$ it means only the features peak-peak, RMS of acceleration values and correlation of acceleration values between pair of axes are considered. We used cumulative misclassification as the index for assessing the best feature set and we used the training set itself for the same. We took the values of N, δ, and Δ as 8, 3, and 2, respectively, for each value of F. For finding the optimal F^*, we placed the main classification window at position of occurrence of each of the movements in the training set and classified each of those movements. The optimal F^* would be the one that gives lower cumulative misclassification. Figure 5(c) shows the cumulative misclassification for various F values and it can inferred that the optimal value of $\mathcal{F} = 7_{10} = [000111]_2$. Hence, the best set of features for classifying movements in a smoking activity includes three features: 1) RMS, 2) Peak-Peak, and 3) Correlation of acceleration values.

B. Optimal Values of Parameters

The performance of the classification algorithm also depends on the N sub-intervals considered within a window for extract features, the number of small classification windows P within the main classification window, and the number of shifts of small classification windows K within the main classification window. To find the optimal value of N, we considered 16 different N values with $\delta = 3$, $\Delta = 2$, and $F = 7_{10}$, which was the optimal value we found out. Figure 5(d) shows the cumulative misclassification ratio for various values of N from which we can say that optimal value of N is around 8. Figure 5(d) shows that if N is too low, the misclassification is higher; also, if N is too high, this would correspond to taking raw acceleration values of the movements. It is also essential to find the optimal values of δ and Δ. Figure 5(e) shows cumulative misclassification ratio vs δ for different Δ, which indicates that the optimal



(a) (b) (c)

Fig. 6. Logical representation of type of movements for Cigarette 1 (a), Cigarette 2 (b), and Cigarette 3 (c).

values of δ and Δ are 3 and 2, respectively, as the cumulative misclassification is the lowest.

C. Number of Accelerometers

There is a tradeoff between cost and comfort of the subject making it essential to identify the minimum number of accelerometers needed to recognize movements without considerably affecting the accuracy. The lower the number of accelerometers, the more comfortable the subject would be. Hence, we used sets of 1, 2, and 3 accelerometers. For 1 accelerometer, we considered two cases of placing the accelerometers: either placing it on the wrist or above the elbow. For the 2 accelerometer set, we placed them on the wrist and above the elbow, while for the 3 accelerometer set, we placed them on the wrist, and below and above the elbow (Table I). Figure 5(f), which depicts the cumulative misclassification ratio vs number of accelerometers, shows that the average cumulative misclassification percentage for the set of 2 accelerometers is slightly lower compared to that of the set of 3 accelerometers. Hence, minimum two accelerometers is enough to classify movements without compromising on the performance. The misclassification percentage is higher when 1 accelerometer is used and is placed above the elbow.

D. Classification Performance

Here we show the classification performance of the algorithm when both the trained and untrained movement sets are from the same subject using 2 accelerometers. Due to lack of space, we show the classification results of only 3 smoking activities. Figure 6 shows the actual and classified positions of both the movements involved in the smoking activity where Movement 1 means ‘arm moving up’ for taking the puff, while Movement 2 means ‘arm moving down’ after smoking. It can inferred from Figs. 6(a-c) that moving average misclassification ratio is around 17% for all the cigarettes. Considering the fact that we did not use any a priori knowledge on the order of occurrence of the movements in the activity, the misclassification rate is acceptable.

V. Conclusions

In this paper, we focused on how effective it would be to use only acceleration data in order to jointly classify the type of movements, and starting and finishing instant of each movement on the fly within a specific activity. The paper also identify the best set of features and the minimum number of accelerometers needed to recognize movements in an activity. The results show accuracy of around 83% when only acceleration data is used to recognize the movements. As future work, we optimize the algorithm further to reduce the cumulative misclassification and also use heterogenous data like using fiber optical sensor to get the angles of motion movement along with accelerometers. We will also implement the movement recognition system using shimmer motes, which provide a more wearable and compact platform.

References

[1] L. Bao and S. S. Intille, “Activity recognition from user-annotated acceleration data,” in *Proc. of Pervasive Computing (PERVASIVE)*, Vienna, Austria, Apr. 2004.

[2] E. A. Heinz, K.-S. Kunze, S. S. H. Junker, P. Lukowicz, and G. Troster, “Experimental Evaluation of Variations in Primary Features Used for Accelerometric Context Recognition,” in *Proc. of European Symposium on Ambient Intelligence (EUSAI)*, Eindhoven, Netherlands, Nov. 2003.

[3] N. Kern, B. Schiele, and A. Schmidt, “Multi-sensor Activity Context Detection for Wearable Computing,” in *Proc. of European Symposium on Ambient Intelligence (EUSAI)*, Eindhoven, Netherlands, Nov. 2003.

[4] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman, “Activity recognition from accelerometer data,” in *Proc. of American Association for Artificial Intelligence (AAAI)*, Pittsburgh, Pennsylvania, Nov. 2005.

[5] H. Ghasemzadeh, J. Barnes, E. Guenterberg, and R. Jafari, “A Phonological Expression for Physical Movement Monitoring in Body Sensor Networks,” in *Proc. of Mobile Ad-hoc and Sensor Systems (MASS)*, Atlanta, GA, Sep. 2008.

[6] M. Muller, T. Rder, and M. Clausen, “Efficient content-based retrieval of motion capture data,” *ACM Trans. Graph.*, vol. 24, no. 3, pp. 677–685, 2005.

[7] L. Feng, Z. Yueting, W. Fei, and P. Yunhe, “3D motion retrieval with motion index tree,” *Comput. Vis. Image Underst.*, vol. 92, no. 2, pp. 265–284, 2003.

[8] J. Barnes and R. Jafari, “Locomotion Monitoring Using Body Sensor Networks,” in *Proc. of Pervasive Technologies Related to Assistive Environments (PETRA)*, Athens, Greece, Jun. 2008.

[9] B. Logan, J. Healey, M. Philipose, E. Tapia, and S. Intille, “A Long-Term Evaluation of Sensing Modalities for Activity Recognition,” in *Proc. of Ubiquitous Computing (UbiComp)*, Innsbruck, Austria, Sep. 2007.

[10] J. Lester, T. Choudhury, and G. Borriello, “A practical approach to recognizing physical activity,” in *Proc. of Pervasive Computing (PERVASIVE)*, Dublin, Ireland, May 2006.

[11] Imote2 datasheet:www.xbow.com.

[12] H. Rangwala and J. Sadagopan, “Support Vector Machines Versus Neural Networks,” *Tech. Rep*, pp. 2004–2043, Mar. 2004.

[13] J. Christopher and C. Burges, “A Tutorial on Support Vector Machines for Pattern Recognition,” *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, Jun. 1998.

[14] Spider:http://www.kyb.mpg.de/bs/people/spider.