Activity Recognition in a Home Setting using Off the Shelf Smart Watch Technology

Avgoustinos Filippoupolitis
Computing and Information Systems
University of Greenwich, UK
Email: a.filippoupolitis@gre.ac.uk

Babak Takand
Computing and Information Systems
University of Greenwich, UK
Email: b.takand@gre.ac.uk

George Loukas
Computing and Information Systems
University of Greenwich, UK
Email: g.loukas@gre.ac.uk

***Abstract*—Being able to detect in real-time the activity performed by a user in a home setting provides highly valuable context. It can allow more effective use of novel technologies in a large variety of applications, from comfort and safety to energy efficiency, remote health monitoring and assisted living. In a home setting, activity recognition has been traditionally studied based on either a large sensor network infrastructure already set up in a home, or a network of wearable sensors attached to various parts of the user's body. We argue that both approaches suffer considerably in terms of practicality and propose instead the use of commercial off-the-shelf smart watches, already owned by the users. We test the feasibility of this approach with two different smart watches of very different capabilities, on a variety of activities performed daily in a domestic environment, from brushing teeth to preparing food. Our experimental results are encouraging, as using standard Support Vector Machine based classification, the accuracy rates range between 88% and 100%, depending on the type of smart watch and the window size chosen for data segmentation.**

1. Introduction

In developing user-centric technologies, it is highly beneficial to know the context, as represented not only by location and time, but also by the activity a user is carrying out. In a home setting, activity recognition can assist in better configured smart home technologies for comfort, security and energy efficiency, and possibly more importantly in remote health monitoring. Especially for the elderly or people with long-term illnesses, activity recognition can help tailor technologies around their evolving needs and even detect signs of worsening of their physical capabilities over time.

Although there are numerous approaches addressing the problem of activity recognition, the majority of them employ purpose built devices to acquire data regarding the user's activities. These may be positioned in multiple parts of the user's body to improve the accuracy of the activity monitoring. Moreover, in a home context, most approaches employ external sensors (usually in a wireless sensor network setting) located inside specific rooms. The goal of this research is to investigate the feasibility of using the onboard sensors of popular commercial off-the-shelf smart watch technology in order to accurately identify activities related to specific rooms or areas inside a home.

To this end, we have developed a framework for data collection and processing that involves wearable devices, a mobile phone and a remote server. Our framework is not tied to a specific proprietary sensor but is able to accommodate a wide range of commercial smart watch devices. This increases the flexibility and adoptability of the approach, since the only requirement for the users is the installation of our mobile application and its pairing with their existing wearable devices.

2. Related Work

In relation to activity recognition in a home environment, most researchers assume the existence of a smart home infrastructure, where a variety of different sensors can be used to detect what the occupants are doing. For example, Chen et al. [1] have experimented with volunteers performing six different activities (cook; watch TV; computer; groom; sleep; bed to toilet) in a three-bedroom smart apartment equipped with 51 motion sensors, in a winter and a summer setting. For classification, the researchers compared Bayes Belief Networks, Artificial Neural Networks, Sequential Minimal Optimisation and LogitBoost Ensemble, with the latter performing consistently better (around 90%) in both settings.

Unlike the above, where the experiments were conducted in a home setting designed specifically for research and had the luxury of the availability of an exceptionally large number of sensors, others have opted for more realistic settings. Kasteren et al. [2] have produced a sensor network setup, which can be easily installed in different houses and is still able to capture the appropriate data for high-accuracy activity recognition. For their evaluation, they focused on eight activities (idle, leaving, toileting, showering, sleeping, breakfast, dinner, drink) and tried to recognise them with hidden Markov models and conditional random fields, both showing promise.

Krishnan and Cook [3] go beyond scripted and pre-segmented activity sequences that are analysed offline, by focusing on real-time recognition based on streaming sensor data. Again the setting is a smart home and the source of

		Home Setting	
		Yes	No
Off the shelf	Wrist		[5], [6]
	Body		[5], [6]
	External		
Experimental	Wrist		[7], [10], [12]
	Body		[8], [9], [10]
	External	[1], [2], [3], [4]	

TABLE 1. COMPARISON OF ACTIVITY RECOGNITION APPROACHES USING OFF THE SHELF AND EXPERIMENTAL TECHNOLOGIES

data is a set of motion sensors installed in the different rooms. For training cost that is acceptable in this type of application, they employed support vector machines as the primary machine learning approach. In addition to realistic activities, one needs to consider also the high likelihood of interleaved activities. Lu and Fu [4] have proposed an activity map to graphically display contextual information about both the occupants and the environment in a sensor-equipped home. In terms of machine learning, they have employed a generalised and enhanced Bayesian Network fusion engine, achieving high accuracy rates for most of the activities.

All the above assume access to a sizeable sensor network infrastructure, which limits considerably the scope and practicality of activity recognition. At the same time, major advances have been achieved over the last few years in the area of activity recognition based on wearable technologies [5], [6], but these have not been evaluated in a home environment (see Table 1 for a summary of the literature, indicating the particular gap). We argue that it is possible to achieve similar accuracy rates for several activities using wearables. For example, Maekawa et al. [7] have developed a wearable device equipped with camera and microphone, while Pirttikangas et al. [8] and Bao et al. [9] have used several accelerometers on different parts of the body, and Mannini et al. [10] have used a single accelerometer placed at the wrist or ankle to capture activities performed by the wearer. However, these apporaches may again be impractical for large-scale adoption. Here, instead of purpose-built sensors attached to the user, we test the feasibility of using the onboard sensors of popular commercial off-the-shelf smart watch technology [11]. Specifically, we perform our evaluations using two very different smartwatches: the richly-equipped MS Band 2 and the more minimal Pebble smartwatch.

3. System Architecture

For the purposes of this project we developed a framework which is modular in nature, and thus can be used for virtually any wearable device which provides an open API. As shown in Figure 1 the system is composed of three essential parts: A wearable device, a smartphone and a server. The reasoning behind this architecture is the optimal use of resources available to each of these sub-systems.

The operation of the system involves a user that is equipped with a wearable device and a mobile device. The user starts the system on their mobile device, which initiates collection of sensor data from the wearable device. Upon a fixed time interval, sensors values are read and transmitted to the user's mobile device via Bluetooth Low Energy. These sensor readings are buffered in the mobile device until the required amount of samples for activity recognition are collected, at which point the data are transmitted to the server over the Internet for processing. The recognition of an activity takes place on the server, which transmits a reply to the mobile device, indicating the recognised activity. Due to the variety of off-the-shelf wearable device platforms, out intention has been to develop a versatile system which can provide a unified solution to any smart device.

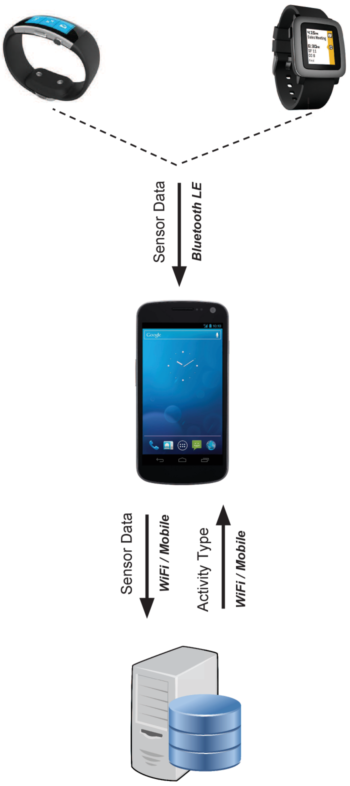

Figure 1. Overall system architecture

3.1. Wearable Devices

Although the concept of wearable computing devices has existed for over a decade, it has been only due to the recent advances in mobile computing that the potential of these devices has started to emerge. A quick search for such devices can reveal a wide variety of platforms, features and capabilities, as well as form factors which largely depend on the targeted usage of such devices.

TABLE 2. OVERVIEW OF SMART WATCH SENSORS

Sensor Type	Band 2	Pebble Time
Accelerometer	✓	✓
Gyrometer	✓	N.E.
Heart Rate monitor	✓	✗
Skin Temperature	✓	✗
Barometer	✓	✗
GPS	✓	✗
Ambient light	✓	N.E.
GSR	✓	✗
UV	✓	✗
Compass	✗	✓

Wearable devices such as smart watches and smart bands are increasingly gaining popularity. Their use so far has been to enhance the use of mobile platforms, and in particular smart-phones. These wearable devices, even in their most basic form, utilise a number of sensors such as accelerometers and light sensors, for several of their essential functions, in a similar way to smartphones and tablets. Moreover, many of these devices are equipped with health monitoring apparatuses which are intended mainly for sport related activities. Table 2 gives an overview of the sensors available on the two commercial smart watch devices we use in our system. Where a particular sensor is physically present in the device but not exposed by the corresponding Application Programming Interface (API), it is marked with N.E.

3.1.1. Pebble. As our initial wearable platform, we have used the Pebble Time smartwatch. Pebble Time was released by *Pebble Technology* in May 2015. In terms of sensors, it features a 3-axis accelerometer, magnetometer, ambient light sensor and gyroscope, although the two latter ones are not at this time exposed in the API, thus no reading can be obtained.

For the collection of sensor data, an application which runs directly on the pebble device was developed. This application runs in the background of the device's operating system, using a "worker". This approach is similar to that of many native applications, such as the pedometer, and notification services. At a frequency of 50 Hz, the background application collects a reading from accelerometer, compass and battery, as well as an epoch timestamp for those readings. These readings are buffered in the internal memory of the pebble, and are periodically pushed to the mobile device, which we have set on one second intervals. Each of these samples forms an array of 20 Bytes. At each transmission to the phone, approximately 50 samples are transmitted.

We note that during the various stages of testing and experimentation the pebble device showed remarkable battery life, as observed in practice. User experience was also very positive in terms of comfort throughout the experiments.

3.1.2. Microsoft Band. The other wearable platform we consider is the Microsoft Band 2. This device, as its name indicates, belongs to the "Smart Band" class of devices, which incorporates a number of smart watch features. As a fitness and health tracker, the Band 2 offers a considerable number of sensors. Specifically, a 3-axis accelerometer, gyrometer, optical heart-rate sensor, galvanic skin response sensor (GSR), ambient light sensor, ultraviolet light exposure sensor and skin temperature sensor.

In our Android application, data collection from the Band is performed via the implementation of event handlers provided for the various sensors of the device in the API. The development of this section has been relatively simpler, as the band does not require a stand-alone application on the device itself. Instead, a number of "sensor events" are provided by the Microsoft Health Service, which is installed alongside the "Microsoft Health App".

The API provides specific sampling rates for the accelerometer and gyroscope. The interval between each sample for these sensors may be either 16, 32 or 128 ms. In our experiments, the sampling interval for the gyroscope and accelerometer was set to 32 ms, which corresponds to a frequency of approximately 31 Hz. Moreover, the Galvanic Skin Response (GSR) sensor has a minimum sampling interval of 200 ms (maximum refresh rate of 5 Hz), while the remaining sensors are updated when there is a change in the last known state of the sensor. In our application, the sample vector is produced on every gyroscope event, thus obtained at a rate of 31Hz.

3.2. Mobile Application

The mobile application, which is developed for the android platform, acts as the keystone of the system. The version presented here, demonstrates the training mode of the application, which is currently used for the collection of training data sets. As part of our unified approach, we use the same app for collecting data from both the Pebble Time, and Band 2 devices. Moreover, the android application follows the modular approach of the system, hence the application can be potentially used for any number of devices with no particular modifications, as for a new device it would be sufficient to develop an introduce a new module which facilitates the integration of a new device's API to our framework, the delivery of which could be done via add-ons to the app.

The current version of the application is intended for collection of sensor data during experiments. For every test subject, an experiment session is created. The capture of sensor data in each session is initiated and terminated though simple start/stop controls on the mobile device. During the experiment data are temporarily stored in the smartphone's internal memory, and are transmitted to the server upon the termination of each session.

3.3. Server

Mobile computing systems, in general, are often characterised by a relatively limited number of resources. Limitations on memory, processing speed, storage, and available power exist in all mobile systems, which only become worse as the size of the computing system decreases. The

TABLE 3. DESCRIPTION OF ACTIVITIES

Activity code	Detailed description
A1	Writing on paper with pen
A2	Typing on keyboard
A3	Stirring soup with spoon
A4	Eating soup with spoon
A5	Cutting with knife
A6	Brushing teeth with toothbrush
A7	Drinking coffee from cup
A8	Washing hands with liquid
A9	Resting hands on laps

computing power provided by wearables such as smart watches and smart bands, allows minimal processing beyond that required for communications, and simple software for notification and data collection. Smartphones offer a significant amount of computational power, although this often depends on the model, make and price range of the device. Moreover, similar to all mobile devices, smartphones have a finite amount of available power. With all these factors in consideration, we incorporate a cloud-based approach to our system, where the actual processing of data, classification and recognition of activities take place on a remote server.

4. Experimental Evaluation

This section elaborates on the experimental methodology we adopted and the experimental results we produced. We first give a description of the method we followed for our experiments and we continue by presenting our classification approach for activity recognition and related results.

4.1. Design and Methodology

Using our framework described in Section 3, we collected data for both smart watches and for a wide range of activities. Table 3 illustrate the activities we included in our experiments.

The rationale behind the selection of activities was to be able to map them (albeit not definitively) in specific rooms/areas of a home. More specifically, the mapping between the activities illustrated in Table 3 and the rooms of a typical home is depicted in Figure 2.

For both smart watches, we used data obtained from their 3-axis accelerometers. (In the case of Band, we have also experimented with both accelerometer and gyrometer data and did not observe a noticeable improvement in performance.) We segmented the data using a non-overlapping sliding window and evaluated our system using different values for the window size (0.5s, 1s, 2s and 3s), as this has been shown to affect the performance of activity recognition [5], [6], [13]. With respect to feature extraction, we used the mean and the standard deviation of the signal for each accelerometer axis. These features were selected based on the analysis in [5]. The data collection was carried out by using our mobile application in training mode. Data coming from the smart watches were sent to the mobile application and then to the server where the processing took place in order to classify the different activities. Each of the aforementioned activities was performed for a time between 160 and 200 seconds and for three different participants. This resulted in a total activities duration of 85 minutes.

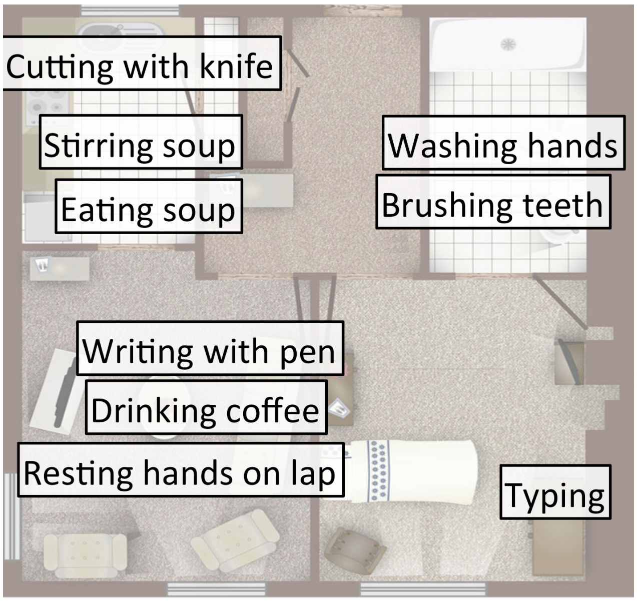

Figure 2. The activities included in the experiments

We partitioned our dataset into 80% training set and 20% test set and used 10-fold cross validation for hyperparameter tuning. We have trained a classifier based on Support Vector Machines with Radial Basis Function kernel (SVM). We selected this type of classifier because it is able to work with non-linearly separable data.

One performance metric for evaluating a classifier is its accuracy, which is defined as the number of correct predictions over the total number of predictions. However, accuracy in general does not optimally define the performance of a classifier since it is negatively affected by class imbalance. Although the methodology we followed to create our dataset inherently minimises class imbalance (since each activity was performed for a similar amount of time), we decided to use a confusion matrix to better describe the performance of each classifier. In our confusion matrix each row represents the instances in an actual class and each column represents the instances in a predicted class. The diagonal elements represent the number of instances for which the predicted label is equal to the true label. Off-diagonal elements represent instances that are misclassified. The higher the diagonal values of the confusion matrix the better. We have chosen to show the confusion matrices normalised by the number of elements in each class. In case of class imbalance, this approach better illustrates which classes are being misclassified. To improve visualisation, we have colour-coded the matrices by assigning black to 1.0 (100%) and white to 0.0 (0%).

4.2. Experimental Results for Band

As we can see from Tables 4-7 the accuracy of the classification depends on the window size. The worse performance is given by the window size of 0.5s. Increasing the window size to 1s improves the classification accuracy for all classes. However, further increasing the window size does not improve the overall system accuracy. More specifically, we can see that for a window size of 1s the classification accuracy of the drinking activity is 95% while the classification accuracy of the teeth brushing activity is 99%. Increasing the window size to 2s results in the drinking activity classification accuracy to increase while that of the brushing activity decreases. In other words, increasing the window size affects the per class accuracy of the classifier but without decreasing the overall accuracy.

	A1	A2	A3	A4	A5	A6	A7	A8	A9
A1	0.98	0	0	0	0	0	0	0.02	0
A2	0	0.97	0	0	0.03	0	0	0	0
A3	0	0	0.98	0	0.01	0	0	0	0
A4	0	0	0.01	0.96	0	0	0.02	0.01	0
A5	0	0.07	0	0	0.91	0	0	0.02	0
A6	0	0	0	0	0	0.96	0.02	0.02	0
A7	0.01	0	0	0.04	0	0	0.95	0	0
A8	0	0	0	0.02	0	0	0	0.97	0
A9	0	0	0	0	0	0	0	0	1

TABLE 4. Confusion matrix for activities A1-A9 (Band, SVM, Window = 0.5 sec)

	A1	A2	A3	A4	A5	A6	A7	A8	A9
A1	0.98	0	0	0	0	0	0.01	0.01	0
A2	0	0.98	0	0	0.02	0	0	0	0
A3	0	0	1	0	0	0	0	0	0
A4	0	0	0	0.98	0	0	0.02	0	0
A5	0	0.01	0	0	0.99	0	0	0	0
A6	0	0	0	0	0	0.99	0.01	0	0
A7	0.03	0	0	0.01	0	0	0.95	0	0.01
A8	0	0	0	0.01	0	0	0.01	0.98	0
A9	0	0	0	0	0	0	0	0	1

TABLE 5. Confusion matrix for activities A1-A9 (Band, SVM, Window = 1 sec)

	A1	A2	A3	A4	A5	A6	A7	A8	A9
A1	0.98	0	0	0	0	0	0.02	0	0
A2	0	0.99	0.01	0	0	0	0	0	0
A3	0	0	1	0	0	0	0	0	0
A4	0	0	0	1	0	0	0	0	0
A5	0	0	0	0	1	0	0	0	0
A6	0	0	0	0	0	0.95	0.03	0.02	0
A7	0	0	0	0	0	0	1	0	0
A8	0	0	0	0	0	0	0	1	0
A9	0	0	0	0	0	0	0	0	1

TABLE 6. Confusion matrix for activities A1-A9 (Band, SVM, Window = 2 sec)

	A1	A2	A3	A4	A5	A6	A7	A8	A9
A1	1	0	0	0	0	0	0	0	0
A2	0	1	0	0	0	0	0	0	0
A3	0	0	1	0	0	0	0	0	0
A4	0	0	0	1	0	0	0	0	0
A5	0	0	0	0	1	0	0	0	0
A6	0	0	0	0	0	0.98	0.02	0	0
A7	0.03	0	0	0	0	0	0.95	0	0.03
A8	0	0	0	0	0.02	0	0	0.98	0
A9	0	0	0	0	0	0	0	0	1

TABLE 7. Confusion matrix for activities A1-A9 (Band, SVM, Window = 3 sec)

4.3. Experimental Results for Pebble

Looking at Tables 8-11 we can see that the classification results in the case of Pebble reveal a different behaviour compared to that of the Band. More specifically, although the window size again affects the accuracy of the classification, we can observe that there is an optimum value for its size. Again the worst performance is given for a window size of 0.5s. Increasing the window size to 1s slightly improves the per class accuracy. For example, the accuracy of the writing activity increases from 95% to 99%. When we use a window of 2s the overall accuracy is further increased. For instance, the drinking activity achieves a 95% accuracy while we are able to achieve 100% accuracy for the writing, cutting, hand-washing and idle activities. If we increase the window size to 3s, the accuracy of the classification is significantly reduced. For example, we can observe that hand-washing now has a 89% accuracy and is mainly mistaken for tooth-brushing. Similarly, writing has a 92% accuracy and is also mainly mistaken for tooth-brushing. As all three activities involve high frequency movements of the hand, increasing the window size impairs the classifiers ability to distinguish among them.

	A1	A2	A3	A4	A5	A6	A7	A8	A9
A1	0.95	0.03	0	0	0	0	0	0.01	0
A2	0.03	0.95	0	0	0.01	0	0	0	0
A3	0	0	0.98	0.01	0	0	0	0.01	0
A4	0.01	0	0.01	0.93	0	0	0.05	0.01	0
A5	0.01	0.04	0	0.01	0.94	0	0	0.01	0
A6	0	0	0	0	0	0.93	0.01	0.06	0
A7	0.01	0.01	0	0.04	0	0	0.91	0.03	0
A8	0.01	0	0	0	0	0.01	0	0.99	0
A9	0	0.01	0	0	0	0.01	0	0	0.98

TABLE 8. Confusion matrix for activities A1-A9 (Pebble, SVM, Window = 0.5 sec)

5. Conclusions and Future Work

In this work, we have designed and developed an activity recognition framework that uses off-the shelf smart watch devices. Our approach is composed of a mobile phone application that gathers data from the smart watch and communicates it to a server where the processing and classification

	A1	A2	A3	A4	A5	A6	A7	A8	A9
A1	0.99	0	0	0	0	0	0	0.01	0
A2	0.02	0.96	0	0	0.01	0	0	0.01	0
A3	0	0.01	0.95	0	0	0	0	0.04	0
A4	0	0	0.01	0.96	0	0	0.02	0.01	0
A5	0.01	0.01	0	0	0.95	0	0	0.02	0
A6	0	0	0	0.01	0	0.93	0.03	0.03	0
A7	0.01	0	0	0.07	0	0	0.88	0.03	0
A8	0.01	0	0	0	0	0.01	0	0.98	0
A9	0	0	0	0	0	0	0	0.01	0.99

TABLE 9. CONFUSION MATRIX FOR ACTIVITIES A1-A9 (PEBBLE, SVM, WINDOW = 1 SEC)

	A1	A2	A3	A4	A5	A6	A7	A8	A9
A1	1	0	0	0	0	0	0	0	0
A2	0	0.98	0	0	0	0	0	0.02	0
A3	0	0.02	0.97	0	0	0.02	0	0	0
A4	0	0	0	0.98	0	0	0	0.02	0
A5	0	0	0	0	1	0	0	0	0
A6	0	0	0	0	0	0.94	0.06	0	0
A7	0	0	0	0	0.02	0	0.95	0.02	0
A8	0	0	0	0	0	0	0	1	0
A9	0	0	0	0	0	0	0	0	1

TABLE 10. CONFUSION MATRIX FOR ACTIVITIES A1-A9 (PEBBLE, SVM, WINDOW = 2 SEC)

	A1	A2	A3	A4	A5	A6	A7	A8	A9
A1	0.92	0	0	0	0	0.05	0	0.03	0
A2	0	1	0	0	0	0	0	0	0
A3	0	0.04	0.96	0	0	0	0	0	0
A4	0	0	0	0.97	0	0	0.03	0	0
A5	0	0.03	0	0	0.94	0.03	0	0	0
A6	0	0	0	0	0	1	0	0	0
A7	0	0.03	0	0	0	0.05	0.92	0	0
A8	0	0	0	0	0	0.11	0	0.89	0
A9	0	0	0	0	0	0	0	0	1

TABLE 11. CONFUSION MATRIX FOR ACTIVITIES A1-A9 (PEBBLE, SVM, WINDOW = 3 SEC)

take place. Our results indicate that even solely with the use of off-the-shelf wearable devices, we are able to accurately recognise a wide range of home activities. However, the results depend on both the smart watch type and on the window size chosen for data segmentation. In future work, we will investigate the performance of our framework with a wider range of commercial off the shelf smart watch technologies, such as Android Wear devices. Furthermore, we will extend our evaluation methodology with additional human activities that occur in a home or other indoor setting. For example, we believe that an extension of the framework in emergency management in indoor areas [14], [15], [16] would benefit the outcome of an emergency operation since it could provide improved situational awareness with respect to building occupant activities (such as kneeling, crawling, door opening) moments before or after an incident. Finally, we will extend the range of machine learning algorithms that we employ and will study the use of neural networks and deep learning. We believe that such methods are worth investigating in view of the importance of efficient human activity recognition.

References

[1] C. Chen, B. Das, and D. J. Cook, “A data mining framework for activity recognition in smart environments,” in *Intelligent Environments (IE), 2010 Sixth International Conference on*. IEEE, 2010, pp. 80–83.

[2] T. Van Kasteren, A. Noulas, G. Englebienne, and B. Kröse, “Accurate activity recognition in a home setting,” in *Proceedings of the 10th international conference on Ubiquitous computing*. ACM, 2008, pp. 1–9.

[3] N. C. Krishnan and D. J. Cook, “Activity recognition on streaming sensor data,” *Pervasive and mobile computing*, vol. 10, pp. 138–154, 2014.

[4] C.-H. Lu and L.-C. Fu, “Robust location-aware activity recognition using wireless sensor network in an attentive home,” *Automation Science and Engineering, IEEE Transactions on*, vol. 6, no. 4, pp. 598–609, 2009.

[5] A. Bulling, U. Blanke, and B. Schiele, “A tutorial on human activity recognition using body-worn inertial sensors,” *ACM Computing Surveys (CSUR)*, vol. 46, no. 3, p. 33, 2014.

[6] O. D. Lara and M. A. Labrador, “A survey on human activity recognition using wearable sensors,” *Communications Surveys & Tutorials, IEEE*, vol. 15, no. 3, pp. 1192–1209, 2013.

[7] T. Maekawa, Y. Yanagisawa, Y. Kishino, K. Ishiguro, K. Kamei, Y. Sakurai, and T. Okadome, “Object-based activity recognition with heterogeneous sensors on wrist,” in *Pervasive Computing*. Springer, 2010, pp. 246–264.

[8] S. Pirttikangas, K. Fujinami, and T. Nakajima, “Feature selection and activity recognition from wearable sensors,” in *Ubiquitous Computing Systems*. Springer, 2006, pp. 516–527.

[9] L. Bao and S. S. Intille, “Activity recognition from user-annotated acceleration data,” in *Pervasive computing*. Springer, 2004, pp. 1–17.

[10] A. Mannini, S. S. Intille, M. Rosenberger, A. M. Sabatini, and W. Haskell, “Activity recognition using a single accelerometer placed at the wrist or ankle,” *Medicine and science in sports and exercise*, vol. 45, no. 11, p. 2193, 2013.

[11] R. Rawassizadeh, B. A. Price, and M. Petre, “Wearables: Has the age of smartwatches finally arrived?” *Communications of the ACM*, vol. 58, no. 1, pp. 45–47, 2015.

[12] U. Maurer, A. Rowe, A. Smailagic, and D. P. Siewiorek, “ewatch: a wearable sensor and notification platform,” in *Wearable and Implantable Body Sensor Networks, 2006. BSN 2006. International Workshop on*. IEEE, 2006, pp. 4–pp.

[13] O. Banos, J.-M. Galvez, M. Damas, H. Pomares, and I. Rojas, “Window size impact in human activity recognition,” *Sensors*, vol. 14, no. 4, pp. 6474–6499, 2014.

[14] A. Filippoupolitis, G. Gorbil, and E. Gelenbe, “Spatial computers for emergency support,” *The Computer Journal*, vol. 56, no. 12, pp. 1399–1416, 2012.

[15] A. Filippoupolitis, G. Loukas, S. Timotheou, N. Dimakis, and E. Gelenbe, “Emergency response systems for disaster management in buildings,” in *Proceedings of the NATO Symposium on C3I for Crisis, Emergency and Consequence Management*. NATO, 2009.

[16] S. Timotheou and G. Loukas, “Autonomous networked robots for the establishment of wireless communication in uncertain emergency response scenarios,” in *Proceedings of the 2009 ACM symposium on Applied Computing*. ACM, 2009, pp. 1171–1175.