EAI Endorsed Transactions
on Context-aware Systems and Applications

Research Article EAI.EU

Towards an Efficient Implementation of Human Activity Recognition for Mobile Devices

Ilham Amezzane[1,]*, Youssef Fakhri[1] Mohamed El Aroussi[1] and Mohamed Bakhouya[2]

[1] Ibn Tofail University, Kenitra, Morocco.
[2] International University of Rabat, Sala Aljadida, Morocco.

Abstract

The availability of diverse and powerful sensors embedded in modern Smartphones/mobile devices has created exciting opportunities for developing context-aware applications. Although there is good capacity for collecting and classifying human activity data with such devices, data pre-processing and model building techniques that achieve this goal are required to operate while meeting hardware resource constraints, particularly for real-time applications. In this paper, we present a comparison study for HAR exploiting feature selection approaches to reduce the computation and training time needed for the discrimination of targeted activities while maintaining significant accuracy. We validated our approach on a publicly available dataset. Results show that Recursive Feature Elimination method combined with Radial Basis Function Support Vector Machine classifier offered the best tradeoff between training time/recognition performance.

Keywords: human activity recognition, smartphone sensors, feature selection.

Received on 08 July 2017, accepted on 09 October 2017 , published on 14 March 2018

doi: 10.4108/eai.14-3-2018.154340

1. Introduction

Researchers are developing many new challenging application scenarios based on mobile phone sensors in various aspects related to the Smart City concept such as in healthcare (e.g., fitness, diabetes, elderly and obesity assisted surveillance), in smart homes (e.g., context aware indoor air quality and thermal comfort control) and in smart transportation (e.g., traffic congestion). Sensors are the necessary elements to make the link between the physical world and the virtual world while the intelligent system performs analysis in the virtual world. It turns out that modern smartphones/mobile devices can play a key role in the recognition of complex states of the user and its environment, particularly the inference of her physical activity thanks to the multitude of embedded sensors and Machine Learning techniques (ML). Human Activity Recognition (HAR) using Smartphones has been widely studied during recent years mainly because Smartphones are not intrusive, widely used in everyday life, and also wearable. In addition, modern Smartphones devices integrate powerful processors, multiple communication technologies, multimedia capability and memory storage. Today's smartphones can also benefit from the growing power of the cloud that allows them to offload expensive computation. Moreover, the variety of available wireless access and communication technologies implemented provide means of long-distance communication to the other user's body wearable sensors. In the following, we will present the smartphone internal sensors used for HAR in our study, then overview the current trends and constraints related to activity recognition process.

*Ilham Amezzane. Email: ilham.amezzane@uit.ac.ma

1.1 Sensors

Accelerometer

Accelerometer has been initially integrated into smartphones in order to automatically determine the orientation in which the user is holding the phone and to adjust the screen to proper viewing when user changes the orientation from landscape/horizontal to portrait/vertical and vice-versa. Accelerometer readings reflect accelerations, i.e. changes in velocity (as the acceleration is the first time derivative of the velocity) and changes in position (by integrating the signal). The accelerometer, however, measures all the accelerations that affect the device, which are the sum of the gravity acceleration and the actual linear acceleration that are associated with the movement of the device on all three physical axis: x (lateral), y (vertical) and z (longitudinal) relative to the screen of the phone as described in Figure 1 (leftside) . The raw data stream from the accelerometer is the acceleration of each axis measured in SI (International System) units, m/s2. A time stamp can also be returned together with the three axes readings. However, some activities are usually indistinguishable using only accelerometer information. Thus, multiple sensors may be required.

Gyroscope

Gyroscope is used to help determine mobile orientation by use of earth‘s gravity. It also adds an additional dimension to the information supplied by the accelerometer by tracking rotation or twist and it is primarily used for navigation and measurement of the angular rotational velocity. Gyroscope measures the phone‘s rotational velocity along the roll, pitch, and yaw axes. The axes directions are shown in Figure 1 (rightside). The raw data from a gyroscope is in rad/s (radian per second). Modern mobile devices refined the gravity measurement by creating a virtual sensor that is implemented as a sensor-fusion of several basic physical sensors, the accelerometer, the gyroscope, and the magnetic sensor instead of the previous low-pass filter which induced an inherent delay. The new technique resulted in much more accurate and fast responding gravity and tilt measurements.

1.2 Activities

Activities recognized by smartphones can be classified in terms of complexity, from simple locomotion such as moving or stationary, walking or running, etc, to more complexe activities such as cooking, dining, brushing teeth, etc. Some applications, use either smartphone embedded sensors or a combination of environmental sensors and smartphone sensors for collecting information, in order to infer context-aware activities. Moreover, since many physical and behavioral characteristics are unique to an individual, smartphone-based personal biometric signature may provide a wide range of possible solutions, which could be used either for the purpose of controlling smart home appliances (e.g., TVs, stereos, dishwashers, heating and light), or to authenticate users and thus to provide an extra level of security and theft prevention.

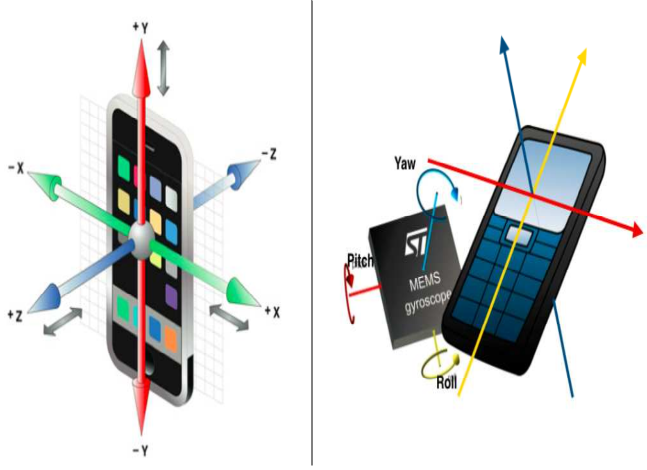

Figure 1: Accelerometer and Gyroscope axes on Smartphones [29, 30]

For example, in [28], Accelerometer-based Personalized Gesture Recognition, called uWave, enables authentication based on physical manipulation of the device with low cost and high efficiency. It is particularly suitable for implementation on resource-constrained devices, such as mobile phones and TV remotes.

A major goal of the current researches in HAR monitoring is to develop new technologies and applications for elderly care. Those applications could help prevent harms, by detecting dangerous situations. For example, it is possible to recognize trends or daily habits of an elderly user, so that it can be easy to early detect any anomaly. The set of activities to be recognized play an important role in the design decisions of an HAR system, because it can help make the appropriate choices regarding sensor selection, feature extraction and classification methods [1].

1.3 HAR Process

Typical HAR can simply be defined as the process of interpreting raw sensor data to classify a set of human activities. ML techniques are used to infer information about the activities from the raw sensor readings; the process consists of the following sequences.

(1) Sensing (or data acquisition): In this step, sensor data are collected at a specific sampling rate from one or more sensors.

(2) Preprocessing: In this step, collected data are processed. For example, noise is removed by applying specific filters and data are segmented.

(3) Feature extraction: In this step, various features are generated. These features are used as inputs to the classifier during recognition phase.

(4) Classification: In this step, the trained classifiers are used to recognize targeted activities.

Before the classification step can be performed for the first time, a model need to be trained. In supervised ML, the training phase requires labeled data to learn the model parameters. The classification phase uses the parameters of the trained model to classify new detected activities. Training can either be offline on a desktop machine or online on the phone itself. In the offline approach, labeled data are first collected and stored. These data are used later for obtaining the model parameters. If training is performed online, the labeled data are directly processed for training and the obtained model parameters are stored for later use in the actual online activity recognition.

1.4 Resource constraints

Although the research on activity recognition is beneficial from the availability and flexibility of smartphone's embedded sensors, it also faces many challenges. Actually, modern Smartphone devices have great capacity for collecting and classifying large amounts of multiple sensor readings. However, energy consumption is of a great a concern, especially for continuous monitoring applications which might deliver critical information. Consequently, in most applications, extending the battery life is a desirable feature. HAR energy consumption is mainly due to sensing, processing, communication, and user interface tasks. By processing, we mean data pre-processing, features extraction, training and classification. Communicating sensor data is generally more energy consuming than processing data locally, therefore, one way of improving battery life of any mobile device is to minimize the amount of transmitted data [31]. Nonetheless, if communicating data is necessary in some applications, short range wireless networks should be preferred over long range networks as the former require lower power. Minimizing communication means more local processing on the device, which means additional computations that may affect the battery life as well. Thus, every step of the HAR process must consider the trade-off between battery life and effectiveness. In the present work, our goal is to create a simple classification model for HAR, by selection of few relevant features without scarifying the performance. This model is intended to be run in online HAR system; therefore, an accelerated implementation is also studied to reduce training time. This is the first step towards energy consumption analysis which will follow in a second phase.

The remainder of this paper is organized as follows. In Section 2 we will present a state of the art of works linked to our goals. Material and methods used in the present study are presented in Section 3. Experimental results are discussed in Section 4. Section 5 presents the conclusions and future work.

2. Related Work

In the literature, many approaches may be used in order to save energy and improve response time during the processing steps, such as adaptive/dynamic sensor selection [8, 9] and adaptive sampling rate [10, 11]. Another important point is whether the classification or training steps should be done online on the device or offline in the server or the cloud. Actually, offline scheme provides more powerful resources capabilities allowing the use of complex approaches for better results, especially if intended for applications that do not require real-time feedback or where online recognition is not necessary because it may introduce additional time to classify activities due to the high computational demands. On the opposite, online scheme should substantially reduce the energy drained from the battery as collected data would not have to be continuously sent to a server or cloud for processing and/or classification. Although it is still a challenging task, online approach would not have been feasible if modern smartphones were not equipped with higher computing and storage resources, even though battery capacities are still scaling more slowly. If online HAR system is implemented fully on-device it offers other advantages such as robustness and responsiveness, thanks to its independence from unreliable wireless communication links or Internet connection that are needed for sending sensor data for further processing to a server or a cloud. Moreover, it avoids transmission of sensitive user information, such as location, activities and user's health. Authors in [1] have reviewed research studies in this domain and stated that only few of them focused on online training in which classifiers can be trained in real time on mobile phones [2-7]. In addition, classifiers could play a key role in HAR process regarding energy consumption depending on their simplicity or complexity. Nonetheless, some of them have proven their suitability for Smartphone implementation, such as K-nearest neighbours (KNN), Support Vector Machine (SVM) and Decision Tree (DT) [1]. Moreover, in the pre-processing phase, various features (or variables) are extracted from sensor readings. They are used later during training or recognition phases. Two main types of features are generally used: time and frequency domain features. It has been shown in [12] that the former are cheaper than the latter in terms of computation and storage costs. However, frequency-domain features may improve classification accuracy. Similarly, large feature sets may significantly slow down the learning process in practice, although extracting more features may improve accuracy [13]. In fact, the "dimensionality curse" phenomenon states that the number of needed training data grows exponentially with the number of dimensions used [14]. Subsequently, the training phase requires further intensive computation if locally undergone on Smartphones [15]. Here come Feature Selection (FS) approaches into play in order to select optimal subsets of variables in the pre-processing step. The main benefits are reducing the computation cost and storage requirements as well as training time [13]. In the following, we review studies done so far that implement online activity recognition systems, especially works that combine the following desirable features:

- They implement online activity recognition locally on the phone.

- They only employ the internal mobile phone sensors, excluding any work that may combine external and internal smartphone sensors.
- Their systems are able to recognize some physical activities.
- Their systems are able to achieve high overall recognition rates
- They should evaluate their systems regarding at least one of the mobile phone resources such as CPU usage/ memory / power consumption.

In particular, we will present the impact of some parameters on mobile phone resources like sensor selection, sampling rate, features set, classification methods and online training.

2.1 Sensor selection impact

Each specific activity may need one or a combination of different sensors for its accurate recognition. However, for a set of various targeted activities, it is possible that some sensors are useful only sometimes during continuous online monitoring for their targeted activities, but still turned on all the time draining power subsequently. Methods such as dynamic and adaptive sensor selection have been used by some researchers in order to improve battery life. This means that unused sensors are turned off in real time in an adaptive way after detection of the mobile status and user's activities. In [8], an adaptive GPS strategy allows the phone to save power by intelligently disabling the GPS as soon as a user enters a building. In [9], authors demonstrated that a hierarchical sensor management strategy significantly improved battery life of the device by selectively turning on the minimum set of sensors and triggering new set of sensors if necessary to achieve state transition detection. It has been shown in the same work that energy consumed by different sensors varies greatly. To further minimize power drain from sensors, some works investigated activity classification using a single accelerometer [33-36]. This approach produces data set that can be smaller than that of a multiple sensor approach, saving power in classification.

2.2 Sampling rate impact

The sampling frequency (SF) has a direct impact on the system's resources. Increasing the sampling rate means more samples and therefore more pre-processing operations and memory usage of the system. In [37], sampling rate of 1 Hz could achieve high performance for detecting simple activities (sitting, standing, walking, and running) in an offline HAR system. Evaluation of the power consumption of the online system with different sampling rates has shown that data processing is the most consuming part, followed by data collection part and then the proposed recognition algorithm part. In their survey, authors in [1] found out that 50, 32 and 20 Hz were the most SF used. Overall, they observe a range from 2 Hz to 125 Hz with reasonable reported accuracies. They also reported that most studies used a fixed SF except two, where adaptive sampling for energy efficiency has been used. In the first one [10], authors used an accelerometer and GPS, where GPS is sampled in an adaptive way. In the other one [11], authors use an activity-sensitive strategy for continuous activity recognition which adaptively makes the choices on both sampling frequency and classification features in real time. Recently, in [38], authors proposed an approach that dynamically controls the activity recognition duration. They investigated the impact of varying the SF and the window size (WS) by starting with a high SF and small WS to quickly identify changing activities. If the same dynamic activity is maintained for a long time, they assume that the same activity will continue and adopt a method to lower the SF and increase the WS. The experimental results showed that their approach reduced energy consumption compared to conventional HAR without sacrificing accuracy. Finally, in [32], an adaptive sampling scheme adjusts the sampling rate of the sensors dynamically based on the context of the user in terms of events observed (i.e., the sensors are sampled at a high rate when interesting events are detected and vice versa) and thus achieves an improved accuracy without considerably compromising on the energy consumption. In [39], authors proposed an adaptive energy-saving strategy by selecting an appropriate combination of flexible frequency and classification feature for each activity. They achieved an overall 28% of energy saving in activity recognition on mobile phone.

2.3 Feature set impact

The longest is the feature vector the better the discerning information it contains for the accurate performance of classification algorithm. However, it takes longer time to collect large data samples for long feature sets. In [40], authors propose a hierarchical method to recognize activities based on a single accelerometer. The proposed method reduces the usage of frequency-domain features and adjusts the WS to improve accuracy. By using SVM and time-domain features, authors in [38] reduced the power consumption by about 6.7% compared to a conventional approach adopting SVM and fast Fourier transform (FFT). They also plot power consumption against different numbers of FFT coefficients. It showed quadratic increase with the dimensionality. Recently, in [41], authors examined the effect of FS techniques on accuracy and ANNs model building time. Obviously this step simplified the model and reduced its time requirement but the recognition rate slightly decreased. For example, with the original feature set the network reached 97% accuracy within 1327s while 95.8% and 619 s with the selected feature set. Some studies combined the effects of more than one parameter. For example, in [42], authors adopt the best sampling rates and extract feature sets in accordance with the different activities, alongside with an improved structure of multi-class SVM combined with the probability of activity occurrence. Experiments show that accuracy is up to 90.6% with 51.0% of energy saving.

2.4 Classification methods impact

Since supervised classification algorithms need intensive computation to generate models from training data, still only few models can be suitable for smartphone implementation. Authors, in [2], introduced “hardware friendly” adaptation for multiclass classification. The proposed approach adapts the standard Support Vector Machine (SVM) and exploits fixed-point arithmetic for computational cost and battery consumption reduction. They highlighted the proportional relationship between the number of bits used and resource consumption. For instance, the 32-bits integer model outperforms in speed and battery life the 32-bit float model. In [43], authors use DT for training all the data which do not spend too much CPU processing and memory, and then they use the SVM to analyze the hard portion only. This way they got the best tradeoff between accuracy and energy consumption. Deep learning approaches are being explored to make them suitable for mobile devices. By extracting features directly from the input data, learning methods can be more generalized instead of hand-crafting appropriate features for each type of activities. However, this is typically achieved by introducing additional layers and nodes for classification, which increases computational complexity. In [44], authors use the Hexagon DSP of the Qualcomm Snapdragon SoC available in some smartphones. This chip is particularly suitable for continuous sensing tasks since it allows the power-hungry CPU to often remain in low-energy sleep mode. They observed on average an 8× to 14× in energy savings when DNNs run on the DSP instead of the CPU. Although, DSP presents several limitations due to its small program and memory space; DNN use is feasible on the DSP and has a low energy and runtime overhead allowing real-time requirements while preserving accuracy. Runtime values of about 16ms are reported for processing 4 seconds of accelerometer data using 24 feature set.

2.5 Online Training impact

In most of the current studies, training is performed offline using ML tools. In fact, the training algorithm which is trained on a large data set of many users have long calculation times, since the training data set increases with every user added to it. In [45] DT, k-NN, and NN have been trained. The faster was k-NN with only 0.1 s. The DT was also fast, with 2.88 s. The training process of the NN was much slower, as it needed 967.16 s. Both the DT and the NN produced a model of ≈1 MB size, k-NN gives a model of ≈10 MB. In [6], authors considered using limited training data which can be collected only in a few minutes considering the limited memory available on the phones. However, there are limitations that can result from using this type of learning process:

- The obtained model is static: once a model is generated it does not adapt to the user's activity profile changes and is not subject specific.
- Computational costs: training batch algorithms usually requires cross-validation techniques in order to avoid overfitting, and that the entire dataset is allocated into main memory.

Subsequently, there is a need towards implementing training models on the mobile devices in an online manner, since online training can help developing fast user dependent applications such as for fitness or healthcare monitoring. In [7], authors demonstrate the Mobile Activity Recognition System (MARS) where for the first time the model is built and continuously updated on-board the mobile device itself using data stream mining. The benefit of data stream mining is that the training samples do not need to be stored on the phone. However, the system will require initial training data for model generation and more data for adaptation. Evaluations of the resource consumption (i.e memory utilization and CPU usage) of the training phase vary proportionally with the data set sizes. Finally, in [46], authors propose a framework for activity recognition called mobile online activity recognition system (MOARS), which integrates data collection, training and inference. The framework dynamically takes into account real-time user feedbacks to increase prediction accuracy, at the cost of additional time. Nonetheless, some activities are recognized from 2 to 4 seconds, depending on the choice of the classifier. It should be noted that no information is provided regarding resource consumption or online training times.

3. Materials and Methods

In this section, we present a comparison study for HAR exploiting feature selection approaches to reduce the computation and training time while maintaining significant performance. Training simulations were also run using parallel resampling of cross-validation over 2 CPU cores.

3.1 Data Assumptions

We use a publicly available dataset from UCI repository [16], which has been used by the authors to conduct experiments using Support Vector Machine (SVM) classifier [17]. The latest update (15-Feb-2015) includes labeled data collected from 30 subjects, within an age bracket of 19-48 years, who engaged in six different activities (standing, sitting, laying down, walking, walking downstairs and walking upstairs), while wearing a Smartphone on the waist. According to the dataset text files in [16], the features selected for this database come from the accelerometer and gyroscope 3-axial raw signals. These time domain signals were captured at a constant rate of 50 Hz. Then they were filtered using a median filter and a 3rd order low pass Butterworth filter with a corner frequency of 20 Hz to remove noise. Similarly, the acceleration signal was then separated into body and gravity acceleration signals using another low pass Butterworth filter with a corner frequency of 0.3 Hz. Subsequently, the body linear acceleration and angular velocity were derived in time to obtain Jerk signals. Also the magnitude of these three-dimensional signals was calculated using the Euclidean norm. Finally FFT was applied to some of these signals. From each window, a vector of 17 signals is then obtained. A set of different statistical variables were estimated from these signals so as each activity window (2,56 seconds) is finally composed of 561-feature vector. The list of all the measures applied to the time and frequency domain signals are also available in [16]. All features were normalized and bounded within [-1, 1]. We observed that the instances in the data set are fairly evenly distributed across the six activity classes. Therefore, the overall accuracy can be representative of the true performance of a classifier. The dataset employed came from a controlled experiment, where subjects did a single activity in a given time frame and the activities were separated with a pause of a few seconds. The experimental results therefore can be attributed solely to the dataset features vector with certainty. Finally, 'subject ID' was not included as a covariate. This is because we do not consider the subject dependency in this dataset. The number and class proportions of observations in each subset are shown in Table 1.

Table 1. Number and class proportions of observations used for training, validation and test.

Dataset	Observations	Class proportions
TRAINING	4694	0.19 : 0.17 : 0.19 : 0.17 : 0.13 : 0.15
VALIDATION	2858	0.18 : 0.17 : 0.17 : 0.15 : 0.13 : 0.14
TEST	2947	0.18 : 0.17 : 0.18 : 0.17 : 0.14 : 0.16

3.2. Feature Selection

In practice, little knowledge is available about features that might capture relevant information to various targeted classes. Therefore, many candidate features are excessively constructed. Given the large feature set, it is expected that many features are irrelevant for any given activity. Actually, these features will represent just noise to the classifier. FS methods aim to select a small subset of features that minimize redundancy and maximize relevance to the target. For example, a feature F_i is said to be relevant to a class C_j if F_i and C_j are highly correlated [18]. For the supervised classification problem, The HAR system operates in two modes: learning (training) from a given set of examples and predicting or (testing). In the training phase, supervised FS may find suitable features that best describes the input patterns guided by the label information. In the classification phase, the system makes automatic decision about unknown input patterns. In literature, there are three main approaches for FS: i) Filter methods for measuring the “relevance” of the features independently of the classifier ii) Wrapper methods for measuring the “usefulness” of the features guided by a classifier performance, and iii) Embedded methods that are implemented by algorithms having their own built-in FS methods for performing variable selection implicitly while the model is being trained. The major advantage of the filter methods is that they are independent of the learning algorithm so that its bias does not interact with the bias of FS algorithm. Indeed, they typically make use of various statistical means to evaluate the relevance of features based on measures of the general characteristics of the training data such as distance, consistency, dependency and information [18]. Therefore, the features can be assigned a weight and they can be ranked. On the other hand, the major advantage of the wrapper models is that they may repeatedly utilize the performance of a learning algorithm to evaluate the usefulness of subsets of selected features. Nonetheless, this method is computationally expensive for data with a large number of features and they have a risk of overfitting to the model. Due to these limitations, the embedded models offer a compromise between filter and wrapper models. Their major advantage is that they perform FS and model fitting simultaneously. In the current study, we examined one method from each category: i) a filter method called Consistency-Based Filter (CBF) ii) a wrapper method called Recursive Feature Elimination (RFE) and iii) an embedded method called Mean Decrease in Accuracy (MDA) built in Random Forest algorithm. The three FS algorithms were run in RStudio [19] using the FSelector package [20] and the Caret package [21, 22] respectively. The advantage of the CBF algorithm is that it includes the effect of interaction among features into evaluation of relevance [23]. Consistency measures, in this regard, evaluate the collective relevance of a set of features and can be considered as a metric to measure a distance of a feature subset from the consistent state [24]. A feature set $\{F_1,..., F_n\}$is said to be consistent, when

$$\Pr(C = c \mid F_1 = f_1,..., F_n = f_n) = 0 \qquad (1)$$

where C is a target class , c is a given class value and f_i is a feature value.

The main drawback of a filter method is that it totally ignores the effect of the selected feature subset on the performance of the learning model [25]. In wrapper approaches, a subset of features is selected and a model is fitted using them. The decision to remove (or add) iteratively features from the subset is taken based on the performance estimation until getting a feature set with the highest performance value. The RFE algorithm consists of a backward greedy search strategy which seems to be particularly computationally advantageous and robust against overfitting [18]. The algorithm fits the model to all features. Each feature is ranked using its importance to the model. Let S be a sequence of ordered numbers which are candidate values for the number of features to retain ($S_1 > S_2$, …). At each iteration of feature selection, the S_i top ranked features are retained, the model is refit and performance is assessed. The value of S_i with the best performance is determined and the top S_i features are used to fit the final model [21]. In our experiments, we tried two learning models with RFE: Random Forest and Linear SVM. Finally, the MDA method is a variable importance algorithm embedded in the Random Forest model. The general idea consists of randomly permuting the values of each feature and measuring how much the permutation decreases the accuracy of the model [26]. If a feature is not useful for predicting a target, then, permuting its values will not result in a significant decrease in a model's performance. The greater the increase in percentage error the greater is the importance of the variable.

3.3. Evaluation

To evaluate whether the feature subsets were more effective for classification than the entire feature set regarding accuracy, training time and computation cost, four common classifiers were run using all features and then using the feature subsets: Linear Discriminative Analysis (LDA), Radial Basis Function Support Vector Machine (RBF SVM), K-nearest neighbors (KNN), and Random Forest (RF). In order to discriminate test data into labeled classes, classifiers have been trained first. The training data have been further automatically divided into 10 smaller datasets for the purpose of 10-fold cross-validation during learning phases. The parameters of the best final models were preserved for testing on holdout data. This classification procedure evaluated the FS results and provided outcome measures to determine which classifier is best suited to online HAR implementation. In general, feature subsets with similar classifier performance to the full feature set should reduce computational burden, thus facilitating on-device implementation.

4. Experimental Results

After running calculations, we obtained 7 features in the subset selected by the CBF algorithm and 20 features in the subset selected by the MDA. The RFE method was tested with 2 wrapper algorithms: RF and Linear SVM. Simulations showed that the highest accuracy estimation was obtained with a 50 feature subset for the former, while it was obtained with a 280 feature subset for the latter. The latter is however not relevant to our comparative study because it showed almost the same accuracy levels as the former in the model generation phase but with longer runtimes. Thus, we excluded the RFE-Linear SVM feature subset in the rest of this paper for the sake of concision. After checking the names and descriptions of the selected features, we noticed that some variables are common between the three subsets and that the CBF subset contains only 2 frequency domain features, while the MDA subset contains 5 and the RFE subset contains 10. This reduction in the number of frequency domain features is beneficial in terms of computation cost, because the original feature set contains many frequency domain features based on FFT, which demands extra computation. Then, we constructed three new datasets with the selected subsets and conducted a comparison study in order to the feature subset that works the best regarding recognition performance and training time. We evaluated our classifiers before and after feature selection, using 10-fold cross validation technique. The parameters of the best final models were preserved for testing on hold-out data.

Because balanced class proportions assumption is verified in our data sets, good performance of each classifier can be measured by accuracy metric only, and good performance of each activity is obtained if that activity can be classified with high precision (PR), recall (RC), and F-measure (F1) metrics.

In Table 2, summary results of the best final models run on the Test data, in terms of time taken to build the model and Accuracy (Acc.). Activities averaged metrics such as, Macro-precision (m-PR), Macro-recall (m-RC) and Macro-F1-measure (m-F1) are also shown. Overall, LDA and SVM offered the highest performance values for the original feature set. It can be seen that the RFE subset offers the best tradeoff performance/training time with the SVM model. For all the other subsets, prediction accuracy is between 87% and 90%. The shortest training time is in favour to KNN. Therefore, recognition phase shows the best stability and performance in favour of SVM for all the subsets. With all subsets, the training time is however far better than with the original feature set.

Because time domain features help preserve battery life by virtue of being less computationally intensive, we discarded all the frequency domain features from the best subset (i.e RFE subset) and the new subset (40 features) was studied using the same procedure as the previous experiments. There was a slight decrease in performance (2~3%) compared to the RFE subset (see Table 2).

TABLE 2: Comparative results: Time/Performance

Sub	***Cl.***	***Total CPU Training time (s)***	***Acc.***	***m-PR***	***m-RC***	***m-F1***
561	SVM	50,71	0.95	0.95	0.95	0.95
	LDA	25,52	0.96	0.96	0.96	0.96
7	SVM	8,6	0.87	0.87	0.86	0.86
50	SVM	7,92	0.92	0.92	0.92	0.92
20	SVM	10,03	0.89	0.89	0.88	0.88
	KNN	2,25	0.87	0.88	0.87	0.87
40	SVM	10,53	0.9	0.89	0.89	0.89
	KNN	2.29	0.89	0.89	0.89	0.89

We also show the F1-measure that for each activity. It can be seen from Table 3, that in all experiments, the “Laying” activity was perfectly classified, followed by Walking”, then “Walking Downstairs”, then “Walking Upstairs”, then “Standing” and finally “Sitting”. The confusion matrix shown in Table 4 contains information about ground truth versus predicted activities of the best SVM final model run on Test data. It can be seen that there is a perfect classification between “moving” and “non-moving” activities.

However, the classifier sometimes confuses and misclassifies one activity from another when there are inter-class similarities. Actually, it confuses little bit between all types of walking activities, but little more between “Sitting” and “Standing”. On the contrary, the “Laying” activity is again perfectly classified. Furthermore, in order to figure out how well they perform over different sized versions of the training set, we have simulated the learning curves of the final SVM models obtained before and after feature selection (see Figure 2). For this purpose, the original Training data (7352 instances) was partitioned into Training set (75%) and Test set (25%), and 10-fold cross validation was used for resampling. As expected, reducing the feature space helped reducing the amount of training data needed to reach the same classification performance. For example, in order to reach 90% of accuracy, approximately 5000 instances were needed before feature selection (Figure 2, Left side), while only 3500 instances were needed after feature selection (Figure 2, Right side). The evaluation of the RFE subset with SVM classifier showed improvements in training time and a slight decrease in activity recognition accuracy compared to the entire feature set. This indicates that the features eliminated in the feature selection process were redundant and did not significantly contribute to classifier accuracy. Thus, with appropriate pre-processing procedure, it is possible to obtain equivalent classifier performance with a smaller feature set, effectively reducing computation burden on the HAR process. It’s worth noting that more than half of the RFE subset (i.e. 27 features) is based on gravity acceleration signals, and 13 features are based on body acceleration signals, while 3 features are based on angles between the XYZ axes and the Gravity Mean signal (see [16] text files for features identification). Hence, for our study, the RFE algorithm found the gravity signals to be more relevant followed by body acceleration signals. The remaining 7 features are based on angular velocity extracted from gyroscope. Statistic estimates such as interquartile range, skewness and kurtosis of the frequency domain signals were not useful. Additional vectors obtained by averaging the signals in a signal window sample used on the angle variable were not useful as well.

Finally, in Table 5, we compare approximations of sequential execution times versus parallel execution times over 2 CPU cores. The values presented are for comparison only. They may vary depending on hardware platforms, the written code, the libraries and the programming languages used. For the sequential part, we notice that total CPU time is close to elapsed time. This means that the codes are CPU bound. For the parallel part, it is clear that it helps reducing CPU time many orders of magnitude. However, still the elapsed time is large in this case. This means that the codes are Memory bound. In other words, more time is spent requesting data than processing it. The reason is that as the number of workers increases, the memory required also increases. For example, using two cores would keep a total of three versions of the data in memory. If the data are large or the computational model is demanding, performance can be affected if the amount of required memory exceeds the physical amount available. In addition, all cores share a single memory bus linking to RAM. Memory Bound would be slower than CPU Bound. Therefore, our ongoing research involves performing training computations using specific framework so that they can be distributed across multiple smartphone devices (CPUs, GPUs or DSPs). This may help alleviate the CPU computation burden for calculations that can be run in parallel and may save more energy as well.

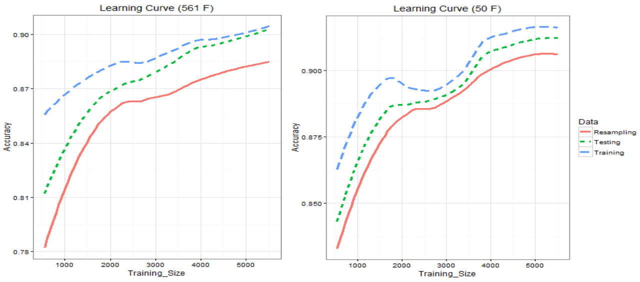

Figure. 2. Learning Curves of the SVM final models obtained: Before feature selection (Left side) and after feature selection (Right side).

TABLE 4: Confusion Matrix of SVM on Test data

Predicted Activities	Ground-truth activities					
	Lay.	Sit.	St.	W.	W-Dn	W-Up
Lay.	537	0	0	0	0	0
Sit.	0	397	35	0	0	0
St.	0	94	497	0	0	0
W.	0	0	0	488	12	37
W- Dn	0	0	0	6	370	10
W- Up	0	0	0	2	38	424

TABLE 5: Comparative results: Sequential vs Parallel runtimes (in seconds)

Sub	Cl.	Sequential		Parallel	
		Total CPU time	Elapsed time	Total CPU time	Elapsed time
561	SVM	1387	1432.5	50.71	944.23
	LDA	229.2	270.88	25.52	153.11
7	SVM	139.1	140.08	8.6	123.24
20	SVM	115.3	122.15	10.03	121.12
	KNN	7.58	7.65	2.25	7.68
40	SVM	146.5	147.9	10.53	136.57
	KNN	14.76	15.73	2.29	14.8
50	SVM	184.6	197.12	7.92	103.6

TABLE 3: Comparative results: F1-measure for each activity

Ac	F1-measure for each activity							
	561		7	50	20		40	
	SVM	LDA	SVM	SVM	SVM	KNN	SVM	KNN
Laying	1.00	1.00	0.99	1.00	1.00	1.00	1.00	1.00
Sitting	0.92	0.92	0.80	0.86	0.79	0.79	0.82	0.80
Standing	0.93	0.93	0.83	0.88	0.82	0.81	0.86	0.85
Walking	0.96	0.98	0.90	0.94	0.93	0.87	0.91	0.92
Walking-downstairs	0.95	0.96	0.86	0.92	0.91	0.90	0.89	0.88
Walking-upstairs	0.95	0.95	0.81	0.90	0.86	0.84	0.86	0.87

4. Conclusion and Perspectives

In this paper, we have conducted a comparison study using Smartphone accelerometer and gyroscope sensors data obtained from a publicly available HAR dataset. As the dimensionality of the original feature set is very high, we used feature selection approaches in order to reduce the feature space before classifying activities. We implemented them in three different experiments in order to evaluate their performance at selecting features in comparison with the first experiment where we used the entire feature set. After analyzing the results, the RFE subset, with only around 9% of the original feature set, showed the best tradeoff between classification accuracy, model building time, and confusion matrix using the SVM classifier. In another experiment, we discarded frequency domain features from the RFE subset to measure their impact. There was a slight little decrease in performance. In general, the SVM classifier offered the best compromise in terms of stability, training time, and performance metrics. This comparison study is our starting point towards finding energy efficient techniques for real time HAR based on Smartphone sensors. Further ongoing research involves an accelerated implementation of the proposed model, which might take advantage of specific hardware computation facilities. Moreover, HAR over smartphones is under development for diabetic control and prediction of hypoglycaemia [27].

References

[1] Shoaib, Muhammad, et al. A survey of online activity recognition using mobile phones. Sensors 15.1 2015: 2059-2085.

[2] Anguita, D.; Ghio, A.; Oneto, L.; Parra, X.; Reyes-Ortiz, J.L. Energy Efficient Smartphone Based Activity Recognition using Fixed-Point Arithmetic. J. UCS 2013, 19, 1295–1314.

[3] Frank, J.; Mannor, S.; Precup, D. Activity Recognition with Mobile Phones. Lect. Notes Comput. Sci. (2011), 6913, 630–633.

[4] Ouchi, K.; Doi, M. Indoor-outdoor Activity Recognition by a Smartphone. In Proceedings of the 2012 ACM Conference on Ubiquitous Computing, Pittsburgh, PA, USA, 5–8 September 2012; pp. 600–601.

[5] Stewart, V.; Ferguson, S.; Peng, J.X.; Rafferty, K. Practical automated activity recognition using standard smartphones. In Proceeedings of the IEEE International Conference on Pervasive Computing and Communications Workshops, Los Alamitos, CA, USA, 19–23 March 2012; pp. 229–234.

[6] Kose, Mustafa.; Incel, O.D.; Ersoy, C. Online Human Activity Recognition on Smart Phones. In Proceeedings of the Workshop on Mobile Sensing: From Smartphones and Wearables to Big Data, Beijing, China, 16 April 2012; pp. 11–15.

[7] Gomes, J.; Krishnaswamy, S.; Gaber, M.; Sousa, P.; Menasalvas, E. MARS: A Personalised Mobile Activity Recognition System. In Proceeedings of the 2012 IEEE 13th International Conference on Mobile Data Management (MDM), Bengaluru, Karnataka, 23–26 July 2012; pp. 316–319.

[8] Ryder,J.;Longstaff,B.;Reddy,S.;Estrin,D. Ambulation: A Tool for Monitoring Mobility Patterns over Time Using Mobile Phones. In Proceedings of the International Conference on Computational Science and Engineering, Vancouver, BC, Canada, 29–31 August 2009; Volume 4, pp. 927–931.

[9] Wang, Y.; Lin, J.; Annavaram, M.; Jacobson, Q.A.; Hong, J.; Krishnamachari, B.; Sadeh, N. A Framework of Energy Efficient Mobile Sensing for Automatic User State Recognition. In Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services, Krakow, Poland, 22–25 June 2009; pp. 179–192.

[10] Lu, H.; Yang, J.; Liu, Z.; Lane, N.D.; Choudhury, T.; Campbell, A.T. The Jigsaw Continuous Sensing Engine for Mobile Phone Applications. In Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems, Zurich, Switzerland, 3–5 November 2010; pp. 71–84.

[11] Yan, Z.; Subbaraju, V.; Chakraborty, D.; Misra, A.; Aberer, K. Energy-Efficient Continuous Activity Recognition on Mobile Phones: An Activity-Adaptive Approach. In Proceedings of the 2012 16th International Symposium on Wearable Computers (ISWC), Newcastle, Australia,18–22 June 2012; pp. 17–24.

[12] Figo, D.; Diniz, P.C.; Ferreira, D.R.; Cardoso, J.M. Preprocessing techniques for context recognition from accelerometer data. Pers. Ubiquitous Comput. (2010), 14, 645–662.

[13] Kotsiantis, Sotiris. Feature selection for machine learning classification problems: a recent overview. Artificial Intelligence Review 2011: 1-20.

[14] Verleysen, Michel, and Damien François. The curse of dimensionality in data mining and time series prediction. International Work-Conference on Artificial Neural Networks. Springer Berlin Heidelberg, 2005.

[15] Bulling, A.; Blanke, U.; Schiele, B. A tutorial on human activity recognition using body-worn inertial sensors. ACM Comput. Surveys (CSUR) 2014, 46.

[16] UCI Machine Learning Repository 2012. Human Activity Recognition Using Smartphones Data Set, http://archive.ics.uci.edu/ml/datasets/ Human+Activity+Recognition+Using+Smartphones

[17] Anguita, D., Ghio, A., Oneto, L., Parra, X., & Reyes-Ortiz, J. L. 2013, April. A Public Domain Dataset for Human Activity Recognition using Smartphones. In ESANN.

[18] Alelyani, S., Liu, H., & Tang, J. 2014. Feature Selection for Classification: A Review. *Data Classification: Algorithms and Applications*.

[19] RStudio Team 2015. RStudio: Integrated Development for R. RStudio, Inc., Boston, MA URL http://www.rstudio.com/.

[20] Romanski, P., Kotthoff, L., & Kotthoff, M. L. (2013). Package 'FSelector'.

[21] Kuhn, M. (2012). Variable selection using the caret package. URL {http://cran. cermin. lipi. go. id/web/packages/caret/vignettes/caretSelection. pdf}. Last accessed 05/31/2017

[22] Kuhn, M. (2012). Variable importance using the caret package.

[23] Shin, Kilho, and Seiya Miyazaki. 2016. "A Fast and Accurate Feature Selection Algorithm Based on Binary Consistency Measure." Computational Intelligence 32 (4): 646–67. doi:10.1111/coin.12072.

[24] Dash, M., Liu, H., & Motoda, H. (2000, April). Consistency based feature selection. In Pacific-Asia Conference on Knowledge Discovery and Data Mining (pp. 98-109). Springer Berlin Heidelberg.

[25] Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. Artificial intelligence, 97(1-2), 273-324.

[26] Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

[27] De Florio, V., Bakhouya, M., Elouadghiri, D., and Blondia, C., Towards a Smarter organization for a Self-servicing Society, 7th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Info-exclusion, pp. 1-6, (2016).

[28] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan, "uwave: Accelerometer-based personalized gesture recognition and its applications," Pervasive and Mobile Computing, 2009.

[29] Apple, UIAcceleration class reference, https://developer. apple.com/library/ios/documentation/uikit/reference/UIAcceleration Class/Reference/UIAcceleration.html, 2014, Mar. 17

[30] Su, X.; Tong, H.; Ji, P. Activity recognition with smartphone sensors. Tsinghua Sci. Technol. 2014, 19, 235–249.]

[31] I. Amezzane, Y. Fakhri, M. El aroussi, M. Bakhouya: FPGA Based Data Processing for Real-time WSN Applications: a synthesis. ICHICS'16, Morocco; 06/2016.

[32] K. K. Rachuri, "Smartphones based social sensing: Adaptive sampling, sensing and computation offloading," Ph.D. dissertation, University of Cambridge, 2012.

[33] Álvarez de la Concepción, M., Soria Morillo, L., Gonzalez-Abril, L., & Ortega Ramírez, J. (2014). Discrete techniques applied to low-energy mobile human activity recognition. A new approach. Expert Systems With Applications, 41(14), 6138-6146. http://dx.doi.org/10.1016/j.eswa.2014.04.018

[34] Guiry, J., van de Ven, P., Nelson, J., Warmerdam, L., & Riper, H. (2014). Activity recognition with smartphone support. Medical Engineering & Physics, 36(6), 670-675. http://dx.doi.org/10.1016/j.medengphy.2014.02.009

[35] Khan, A., Siddiqi, M., & Lee, S. (2013). Exploratory Data Analysis of Acceleration Signals to Select Light-weight and Accurate Features for Real-time Activity Recognition on Smartphones. Sensors, 13(10), 13099-13122. http://dx.doi.org/10.3390/s131013099

[36] Martín, H., Bernardos, A., Iglesias, J., & Casar, J. (2012). Activity logging using lightweight classification techniques in mobile devices. Pers Ubiquit Comput, 17(4), 675-695. http://dx.doi.org/10.1007/s00779-012-0515-4

[37] Weng, S.; Xiang, L.; Tang, W.; Yang, H.; Zhneg, L.; Lu, H.; Zhneg, H. A Low Power and High Accuracy MEMS Sensor Based Activity Recognition Algorithm. In Proceedings of the 2014 IEEE International Conference on Bioinformatics and Biomedicine, Belfast, UK, 2–5 November 2015; pp. 33–38.

[38] Jin Lee and Jungsun Kim, "Energy-Efficient Real-Time Human Activity Recognition on Smart Mobile Devices," Mobile Information Systems, vol. 2016, Article ID 2316757, 12 pages, 2016. doi:10.1155/2016/2316757

[39] Viet, V. Q., Thang, H. M., Choi, D. Adaptive energy-saving strategy for activity recognition on mobile phone. Proceeding of 12th IEEE International Symposium on Signal Processing and Information Technology2012

[40] Y. Liang, X. Zhou, Z. Yu, and B. Guo, Energy-efficient motion related activity recognition on mobile devices for pervasive healthcare. Mobile Networks and Applications, pp. 1-15, 2013

[41] Suto, J. & Oniga, S. J Ambient Intell Human Comput (2017). https://doi.org/10.1007/s12652-017-0513-5

[42] Chen, Z., Wu, J., Castiglione, A., and Wu, W. (2016) Human continuous activity recognition based on energy-efficient schemes considering cloud security technology. Security Comm. Networks, 9: 3585–3601. doi: 10.1002/sec.1563.

[43] Wei Zheng, Yuri Yoshihara, Tay Noel, Dalai Tang, Naoyuki Kubota. Energy-efficient Activity Recognition on Smartphone. 2016 Third International Conference on Computing Measurement Control and Sensor Network. DOI: 10.1109/CMCSN.2016.32

[44] N. D. Lane and P. Georgiev, "Can deep learning revolutionize mobile sensing?" in International Workshop on Mobile Computing Systems and Applications. ACM, 2015, pp. 117–122.

[45] Susanna Spinsante, Alberto Angelici, Jens Lundström, Macarena Espinilla, Ian Cleland, and Christopher Nugent, "A Mobile Application for Easy Design and Testing of Algorithms to Monitor Physical Activity in the Workplace," Mobile Information Systems, vol. 2016, Article ID 5126816, 17 pages, 2016. doi:10.1155/2016/5126816

[46] Lu, Dang-Nhac, et al. "Mobile Online Activity Recognition System Based on Smartphone Sensors." International Conference on Advances in Information and Communication Technology. Springer International Publishing, 2016.