Human Activity Recognition for Indoor Positioning using Smartphone Accelerometer

Sara Khalifa[1 3] Mahbub Hassan[1 3] Aruna Seneviratne[2 3]

[1] School of Computer Science and Engineering,
University of New South Wales, Australia
{sarak,mahbub}@cse.unsw.edu.au
[2] School of Electrical and Telecommunication Engineering,
University of New South Wales, Australia
a.seneviratne@unsw.edu.au
[3] National ICT Australia, Locked Bag 9013, Alexandria, NSW 1435, Australia
{sara.khalifa, mahbub.hassan, aruna.seneviratne}@nicta.com.au

Technical Report
UNSW-CSE-TR-201303
January 2013

THE UNIVERSITY OF
NEW SOUTH WALES

School of Computer Science and Engineering
The University of New South Wales
Sydney 2052, Australia

Abstract

With indoor maps showing facility locations, the activity context of the user, such as riding an escalator, could be used to determine user position in the map without any external aid. Human activity recognition (HAR), therefore, could become a potential aid for indoor positioning. In this paper, we propose to use the smartphone accelerometer for HAR of two key indoor positioning activities, riding an escalator (E) and riding a lift (L). However, since users do not actually perform any specific physical activity during E and L (they typically stand still in escalator or lift), HAR of these two activities is a challenging problem. We conjecture that the smartphone accelerometer would capture the characteristic vibrations of escalators and lifts, making it possible to distinguish them from each other with reasonable accuracy. We collect a total of 177 accelerometer traces from different individuals riding different lifts and escalators in different indoor complexes under natural conditions, and apply different combinations of noise filtering, feature selection, and classification algorithms to these traces. We find that using only the raw accelerometer data, the E and L activities can be recognized with 90% accuracy, but a simple moving average filter would increase the accuracy to 97%. We, however, discover that a third indoor activity, standing still on the floor (S), which could be confused with E and L, reduces recognition accuracy noticeably from 97% to 94% for the filtered data. An interesting finding is that the moving average filter leads to simpler features for classification, which may ultimately compensate for any increase in HAR overhead due to filtering.

1 Introduction

To assist visitors finding the location of various facilities, such as lifts and escalators, authorities of large indoor complexes publicly release the floor maps on their web sites. See for example (Fig. 1.1), the map of the second floor of Sydney international airport, which shows the locations of many different types of facilities, including lifts, escalators, and stairs. A user can preload these maps to the smartphone before visiting the indoor complexes. Conceptually, it is then possible to determine indoor positioning from the activity context of the user. For example, if the user is detected to be riding an escalator, then her position can be narrowed down to discrete locations on the map. Human activity recognition (HAR) in a smartphone, therefore, could become a potential aid for indoor positioning. Indeed, HAR has been recently used [15] to reduce positioning errors of traditional pedestrian dead reckoning based indoor positioning.

One of the other advantages of using HAR with preloaded maps is privacy. As all computations are done locally using only local sensor data, there is absolutely no leakage of private information, such as WiFi association fingerprints [14]. Thus it complements the current efforts in privacy preserving techniques [24].

However, to realize the positioning potential of HAR, we have to address several practical issues. First, the HAR overhead has to be minimal, so continuous use of it does not drain the limited battery of the smartphone. This requires that we have to be economical not only about our selection of sensors, but also the HAR algorithms that make use of sensor data. Sources of HAR algorithmic overhead include the number and type of features to be extracted from sensor output for the activity classification and any filtering used to process raw sensor data before the classification.

Second, the HAR models should be independent of users and environments, i.e., they should work effectively for any user and any indoor complex and facilities. Third, the HAR should work for natural holding positions of the smartphones, i.e., there is little practical use if the smartphone has to be strapped to specific body parts for accurate activity recognition. Finally, good recognition accuracy must be achieved for key indoor positioning activities, because inaccurate recognition misleads the map-matching-based indoor positioning system.

In this paper, we propose to use only the smartphone accelerometer for HAR of typical indoor positioning activities. Accelerometer consumes much less power [37] [10] than most other sensors, such as radio (WiFi, Bluetooth, 3G, etc.), audio (microphone), and image/video (camera). Since the smartphone uses accelerometer data for automatic screen layout adaptation (portrait vs landscape), it is always turned on anyway during the use of the smartphone making additional power consumption due to HAR negligible.

We focus on two indoor positioning activities, riding an escalator (E) and riding a lift (L), because they are universally included in the publicly available indoor maps (see Fig. 1.1). However, since users do not actually perform any specific physical activity during E and L (they typically stand still while using these facilities), reliably distinguishing these two activities from each other using only accelerometer data is a challenging problem. We, however, observe that lifts and escalators could have their own subtle characteristic vibration patterns due to riding quality standards specifying different tolerance patterns for

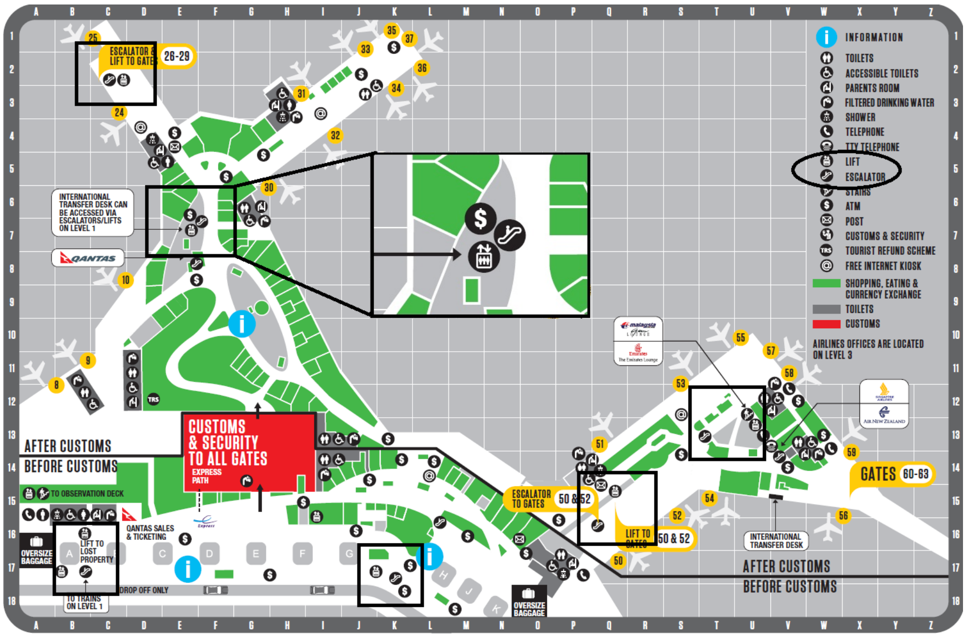


Figure 1.1: Floor Map of Level 2 in Sydney Airport [6]. The map shows locations of 9 lifts and 9 escalators together with many other facilities

horizontal and vertical vibrations of such facilities [3] [21]. It is, therefore, our conjecture that the smartphone accelerometer would capture these vibration patterns, which could be the basis for classifying these activities with appropriate machine learning tools.

To investigate the validity of our conjecture, we collect accelerometer traces from different individuals riding different lifts and escalators in different indoor complexes under natural (non-laboratory) conditions. We then apply different combinations of noise filtering, feature selection, and classification algorithms to these traces. We find that using only the raw accelerometer data, the E and L activities can be recognized with 90% accuracy, but a simple moving average filter would increase the accuracy to 97%.

Standing still on the floor (S) is another typical indoor activity that does not contribute to positioning, but can potentially be confused with E or L. Such confusion is harmful, because it may cause significant positioning error, e.g., a user standing away from an escalator may be matched to the nearest escalator. We therefore collect traces for the S activity as well from different subjects and retrain our classifiers for the three activities together. We find that S reduces recognition accuracy from 97% to 94% for the filtered data, but has no noticeable effect when data is not filtered.

The novelty and contributions of this paper can be summarized as:

- We collected accelerometer data for three different indoor activities, E, L, and S. Our data collection involved a total of five subjects, 18 different escalators, and 11 different lifts from nine different indoor complexes giving a total of 177 accelerometer traces. All data were collected under natural non-laboratory conditions.

- Using these traces, we compare the performance of three different feature reduction algorithms, *Information Gain*, *Correlation-based Feature Selection*, and *Decision Tree Pruning*. We find that Decision Tree Pruning performs the best and can drastically reduce the number of features needed for HAR without compromising accuracy.
- We compare five different classifiers, Decision Table (DTL), Decision Tree (DT), Naïve Bayes (NB), K Nearest Neighbour (KNN), and Multilayer Perceptron (MLP). We find that although DT outperforms all other classifiers when only E and L are considered, MLP yields the best accuracy if S is introduced to the set of activities.
- We show that application of a simple moving average filter on the raw acceleration data increases recognition performance significantly. An interesting finding is that the moving average filter leads to simpler features for classification, which may ultimately compensate for any increase in HAR overhead due to filtering.
- We find that there is a 12% probability that an S activity may be incorrectly recognized as an E or L.

The rest of the paper is organized as follows. Related work is reviewed in Section 2. We explain the data collection process in Section 3, followed by the HAR methodology in Section 4. Results are presented and analyzed in Section 5. We conclude the paper in Section 6 with a discussion of future work.

2 Related Work

Human activity recognition (HAR) has been an area of significant research in the literature over the past years. All the approaches for HAR share 3 basic components, data collection, feature extraction, and classification. In the data collection phase, most of the published work relied on attaching accelerometer sensors to different places on the human body (wearable sensors). However, the popularity of smartphones in the past few years has shifted the research attention to use these devices for HAR. Table 2.1 summarizes most of the studies related to performing HAR using either wearable sensors or smartphones, in terms of the position of the device on the user's body, E and/or L activities are included or not, the number and description of the features, and the classifier(s) used.

Table 2.1 does not provide a simple comparison between the different studies, since every research is applied on different dataset and considered different type of activities. It might be more difficult to distinguish two similar activities than distinguishing a large number of dissimilar activities. However, this table presents an overall view of the basic framework used in the related studies. We note from Table 2.1 that, although the number of the basic features used seems small (on average 6), but the exact number of features to be fed to the classifier is very large (up to 75) [13] which increases the HAR overhead not only in terms of the computational time needed to calculate all of these features but also the complexity of the classifier that will use all of these features to classify the activities. We also note that the number of features used for classification is not the only important factor that imposes overhead for HAR, but also the

Table 2.1: Summary of some prior works on accelerometer-based HAR (Note: Feat. No. means the exact number of features to be fed to the classifier)

Device	Ref.	Position of the device	E and/or L included?	Feat. No.	Features	Classifiers
Wearable Sensors	[13]	Five 2-axis accelerometers (different places)	Both	75	Mean, FT Energy, Frequency Domain Entropy, Correlation	DTL, KNN , DT , NB
	[31]	One 3-axis accelerometer(near the pelvic region)	Neither E nor L	12	Mean, Standard Deviation, FT Energy, Correlation	DTL, DT, KNN, SVM, NB
	[12]	One 2-axis accelerometer (waist)	Neither E nor L	10	Mean, Standard Deviation, Skewness, Kurtosis, Eccentricity	MLP
	[9]	Five 3-axis accelerometer (different places)	L only	30	Mean, Variance, Skewness, Kurtosis, Autocorrelation, The Peaks of the DFT	BDM, RBL,LSM , KNN, DTW, ANN, SVM
Smartphone	[20]	In the pocket of the front pants leg	Neither E nor L	43	Average, Standard Deviation, Average Absolute Difference, Average Resultant Acceleration, Time Between Peaks, Binned Distribution	DT, LR, MLP
	[15]	In the user hand in front of the body	L only	25	Velocity, Distance, Mean, Variance, Standard deviation, Interquartile Range, Root Mean Square, Correlation	SVM
	[28]	Strapped to the user's ankle	L only	4	Mean, Variance, Skewness, Kurtosis, Eccentricity, Correlation	NB, DTW
	[11]	In the right palm of the hand with the screen faced upwards	Neither E nor L	6	Mean, Standard Deviation	NB

Abbreviations: Discrete Fourier Transform (DFT), Decision Table (DTL), Decision Tree (DT), Naïve Bayes (NB), K Nearest Neighbour (KNN), Multilayer Perceptron (MLP), Support Vector Machine (SVM), Logistic Regression (LR), Bayesian Decision Making (BDM), Rule Based Learner (RBL), Least square method (LSM), Dynamic Time Wrapping (DTW), Artificial Neural Network (ANN).

(a) Escalator samples collection (from Centro Bankstown Shopping Center)

(b) Lift samples collection (from CSE building in UNSW)

Figure 3.1: Smartphone holding position (a) Escalator samples collection and (b) Lift samples collection

type of these features. For example, features extracted from the frequency domain increase the computational burden and impose additional complexities to storage (since the signal has to pass through Fourier Transform).

References [28] and [11] use simple (time domain) and small number of extracted features. However, in [28], the authors strapped the phone to the user's ankle to keep the y-axis of the phone aligned to the lower leg at all times. Therefore, the activities have had distinguishable characteristics in the accelerometer data. This distinction in the signals have made the classification process, to some extent, an easy job and has allowed the authors to rely on only 4 features. In [11], the chosen activities were dissimilar (sitting, standing, walking, running and jumping) and hence were easy to be distinguished using small number of features (6 features).

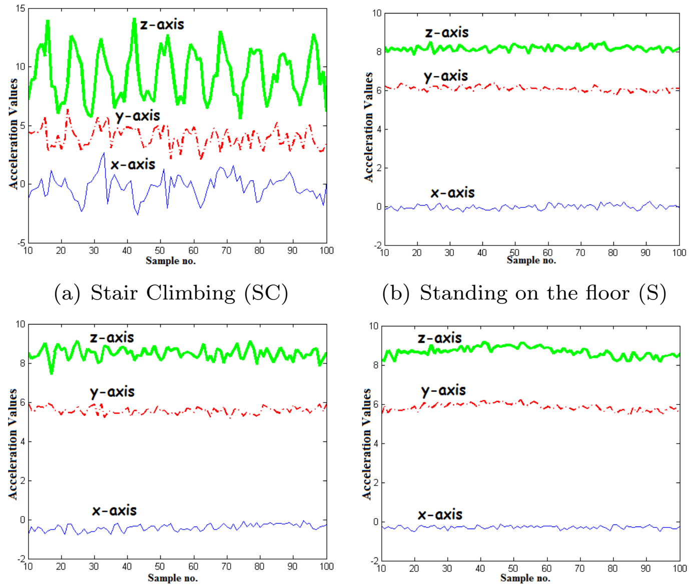
It can be seen, from Table 2.1 that, the activities studied in most of the surveyed papers either did not include both L and E or include L but not E. It is also worth noting that the single paper which included both E and L [13] reported a poor recognition accuracy (70.56% for E), and (43.58 % for L) in spite of using a large number of sensors attached to different positions of the user's body and the high number of features used (up to 75 features some of which are extracted from the frequency domain).

3 Data Collection

Our data were collected using an Android Galaxy Nexus smartphone and a publicly available accelerometer data collection software called AccelerometerValues [1]. Once activated, AccelerometerValues simply records the x, y, and z axes values of the accelerometer at a specified frequency. We configured it to collect data at 20Hz, which records one 3-dimensional acceleration reading every 50 ms.

3.1 Raw Data

The data is collected from nine different indoor complexes, National ICT Australia (NICTA) building [4] (2 lifts), four buildings in UNSW: Computer Science and Engineering (2 lifts), Mechanical Engineering (2 lifts), Electrical Engineer-



(a) Stair Climbing (SC) (b) Standing on the floor (S)

(c) Riding Escalator (E) (d) Riding Lift (L)

Figure 3.2: Samples of four different activities (a) Stair Climbing, (b) Standing on the floor, (c) Riding Escalator and (d) Riding Lift

ing (3 lifts), and Graduate Research School (1 lift), 2 shopping centers: Parramatta Westfield [5] (10 escalators), Centro Bankstown [2] (6 escalators), and 2 train stations: Redfern train station (2 escalators), Belmore train station (1 lift), consisting of a total of 18 different escalators and 11 different lifts. Five volunteers, 3 males and 2 females of ages between 25 and 35, were asked to hold the smartphone on the right or left palm of the hand in front of the body[1] as shown in Fig. 3.1 and perform the three specified activities, S, L, and E. While riding lift or escalator, the subjects were told to simply stand on the moving platform and not walk around or climb up or down.

For escalators, data collection begins and ends at two end points of the escalator, giving a trace length proportional to the length of the escalator. For most of the escalators, the traces were about $20sec$ long, with the exception of Redfern Train Station, where escalators were much longer ($40sec$). For lifts, it is harder to control the trace length as lifts are stopped arbitrarily by other users in the building. Therefore, our lift trace lengths varied widely ranging from a mere $5sec$ (one floor) to $20sec$ (5 floors). To match the majority of traces, all S activity traces are $20sec$ long. From five subjects, we collected a total of 177 traces, including 64 E's, 80 L's, and 33 S's. With a $20Hz$ data collection frequency, we have 20 3D data for each second of the trace.

Stair climbing (SC) is another activity that could be used for indoor positioning. However, with SC, the users periodically move their legs, which would produce clear patterns in the accelerometer signals, making it easily distinguishable from E and L. To verify this, we have collected some limited number of SC

[1]This is the most natural holding position when using the phone for applications that may need positioning information.

traces. Fig. 3.2 shows 4 samples, one from each of the four activities, E, L, S, and SC. It is clear that SC can be readily distinguished from the other three activities, but the differences between E, L, and S are not obvious. Therefore, in this paper, we focused on distinguishing E, L and S activities from each other.

3.2 Data Filtering

Because our data were collected from natural settings outside the lab, the raw data could contain noise caused by various sources, including unexpected movement of the subjects and platform vibrations caused by other people riding the same lift or escalator. Such noise could reduce HAR accuracy and (or) increase the number of features needed for classification. It is therefore appropriate to consider filtering the raw acceleration data before using them for HAR. Past works have used complex filters, such as Butterworth low-pass filter and discrete wavelet package shrinkage [36]. To apply such filters, however, transformations such as Fourier Transform and Wavelet Transform need to be applied to the signal first, which would increase the computational burden for a smartphone. In this paper, we propose to use a moving average filter (MAF), which is very simple yet is optimal for removing random noise from time series while retaining a sharp step response [32]. MAF is calculated by averaging a number of points from the input signal specified by a window (we used a 3-point window) to produce each point in the output signal. We applied MAF to every one of the 177 collected traces, generating two sets of data, one containing the raw data and the other filtered. The two sets are used separately for feature extraction, feature reduction, and classification as explained in the following sections.

4 Activity Recognition Methodology

Identifying relevant features from the accelerometer data for classifying target activities with good accuracy and minimal overhead is the main objective of the proposed HAR. In this section, we present our methodology for feature extraction, feature reduction, and classification.

4.1 Feature Extraction

From the many features used in the literature for accelerometer-based HAR (see Table 2.1), we chose 7 features that are all computed easily in time-domain (no frequency-domain transformation of the signal is required). These are shown in Table 4.1, the number of features generated for each feature-type is shown in brackets. Note that the first 5 features are single-dimension features and we compute the feature for each of the three axes, giving a total of 15 extracted features. The 6^{th} feature is a correlation between two axes, hence we have a total three of these, Corr(x,y), Corr(x,z), and Corr(y,z). Finally, the 7^{th} feature is computed once using all three axes, hence giving just one feature computation of this type. Overall, we extract a total of 19 features for each accelerometer trace.

Our traces are of variable length containing as few as 150 samples to 750 samples, where each sample is a 3D reading of the accelerometer. For each of the 177 traces, we divide the entire trace into a few non-overlapping *windows*

Table 4.1: The initial feature set before applying feature reduction (seven basic features yielding a total of 19 extracted features)

Feature Name (No. of features)	Equation
Mean (3)	$\mu(x) = \frac{1}{n}\sum_{i=1}^{n} x_i$
Standard deviation (Std) (3)	$\sigma(x) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2}$
Skewness (3)	$Skew(x) = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)^3}{\sigma^3}$
Kurtosis (3)	$Kurt(x) = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)^4}{\sigma^4} - 3$
Average Absolute Deviation (3)	$AAD(x) = \frac{1}{n}\sum_{i=1}^{n} \lvert x_i - \mu \rvert$
Correlation (3)	$Corr(x, y) = \frac{Cov(x,y)}{\sigma(x)\sigma(y)}$
Average Resultant Acceleration (1)	$ARA(x, y, z) = \frac{1}{n}\sum_{i=1}^{n}\sqrt{x_i^2 + y_i^2 + z_i^2}$

The "minus 3" in the kurtosis equation, is often used as a correction to make the kurtosis of the normal distribution equal to zero.

each 100-sample long[1]. The number of windows in a given trace therefore equals to $\left\lfloor \frac{TraceLength}{100} \right\rfloor$. For each window, we extract 19 features (see table 4.1, but the feature values across all windows of a given trace are averaged to represent the final feature values for that trace. Thus, after the feature extraction, we have a 177x19 feature matrix giving 19 feature values for 177 traces. We repeat this process for the filtered data, where we consider 177 filtered traces.

4.2 Feature Reduction

Since the target application platform is a smartphone, our objective is to use a minimum number of features for the activity classification without sacrificing recognition accuracy. To this end, we employ two classical techniques for feature reduction [35], Information Gain (IG) and Correlation Feature Selection (CFS). A third technique, called Decision Tree Pruning (DTP), which is obtained as a byproduct of a particular type of classification (explained in the following subsection), is also considered and compared against the other two. Note that all these feature reduction techniques, as well as the classification techniques mentioned in the following subsection, are implemented in the widely used WEKA [7] [34] software, which we used for our study. In this subsection, we explain the outcome of IG and CFS:

- Information Gain: IG is usually used in decision tree analysis to select the candidate feature for branching at each step while growing the tree [23]. The IG of feature f_i measures the expected reduction in entropy caused by partitioning the examples according to this feature. The calculation

[1]For a sampling rate of 20Hz, this corresponds to 5 seconds, which has been found to be sufficient to detect a human activity [9].

of information gain is based calculating the entropy of a set of features S, from:

$$H(S) = -\sum_{i=1}^{n} p_i \log_2 p_i \tag{4.1}$$

where n is the number of different activity classes and p_i is the proportion of all traces belonging to the i^{th} class. The information gain is then calculated using:

$$Gain(S, f_i) = H(S) - \sum_{v \in Values(f_i)} \frac{|S_v|}{|S|} H(S_v) \tag{4.2}$$

where S_v is the subset of S for which feature f_i has a value v (i.e., $S_v = s \in S | Values(f_i) = v$) and $|S|$ denotes the cardinality of the set S.

Table 4.2 shows the results of applying this technique to the raw and filtered data for both scenarios, without and with the S activity. It is interesting to note that the rankings are different in different scenarios, but there is always some features with zero information gain. For each scenario, a reduced set of features is obtained by discarding the features having a zero gain. Table 4.2 shows that: When only the activities E and L were included, the 19 features were reduced to 11 for the raw data and to 17 for the filtered data. When activity S is added to the activities L and E, the 19 features were reduced to 14 for the raw data and to 16 for the filtered data. It is clear that the IG technique had no significant impact in reducing the number of features when the filtered data were used.

- Correlation Feature Selection: The set of n features is partitioned to subsets of size k, $1 \leq k \leq n$, CFS then evaluates the worth (or merit) of each subset of features, M_S, using:

$$M_S = \frac{k\bar{r_{cf}}}{\sqrt{k + k(k-1)\bar{r_{ff}}}} \tag{4.3}$$

where, $\bar{r_{cf}}$ is the mean feature-class correlation ($f \in S$), and $\bar{r_{ff}}$ is the average feature-feature inter-correlation. The merit score takes into account the usefulness of individual features for predicting the class label. Broadly speaking, feature subsets with high average correlation to the class and low inter-correlation receive higher merit scores. CFS, then acts as a simple filter algorithm that ranks feature subsets having a reasonably high merit, according to a search strategy based on a correlation evaluation function, and the optimal subset is the one that satisfies the CFS criterion in:

$$CFS = \max_{S_k} \left[\frac{r_{cf_1} + r_{cf_2} + \ldots + r_{cf_k}}{\sqrt{k + 2(r_{f_1 f_2} + \ldots + r_{f_i f_j} + \ldots + r_{f_k f_1})}} \right] \tag{4.4}$$

where the Pearson's correlation coefficient is used for calculating the correlation between pairs of features ($r_{f_i f_j}$) and Spearman's correlation coefficient for each feature with the target (r_{cf_i}). More details about CFS are given in [16].

Table 4.2: Feature Reduction with Information Gain (IG)

E and L				E, L, and S			
Raw Data		Filtered Data		Raw Data		Filtered Data	
Feature	Gain	Feature	Gain	Feature	Gain	Feature	Gain
Corr(y,z)	0.39	Std(z)	0.76	AAD(z)	0.73	AAD(z)	0.99
AAD(x)	0.21	AAD(z)	0.68	Std(z)	0.70	Std(z)	0.73
Std(x)	0.20	Corr(y,z)	0.66	Corr(y,z)	0.53	Corr(y,z)	0.64
ARA(x,y,z)	0.19	Kurt(y)	0.33	AAD(y)	0.39	Kurt(y)	0.46
Kurt(z)	0.19	Skew(z)	0.31	Std(y)	0.39	AAD(y)	0.40
Mean(z)	0.18	Skew(y)	0.30	ARA(x,y,z)	.22	Skew(y)	0.33
Mean(y)	0.17	Std(y)	0.23	Corr(x,y)	0.21	Kurt(z)	0.29
Corr(x,y)	0.12	Kurt(z)	0.18	Mean(y)	0.20	Skew(z)	0.28
Skew(z)	0.08	Mean(y)	0.18	Std(x)	0.20	Std(y)	0.27
AAD(z)	0.82	Mean(z)	0.16	AAD(x)	0.20	Mean(y)	0.20
Std(z)	0.08	AAD(y)	0.16	Mean(z)	0.20	Mean(z)	0.19
Mean(x)	0	AAD(x)	0.13	Kurt(z)	0.15	AAD(x)	0.12
Std(y)	0	Std(x)	0.12	Corr(x,z)	0.12	Corr(x,y)	0.11
Kurt(y)	0	Corr(x,y)	0.11	Skew(z)	0.07	Std(x)	0.11
Corr(x,z)	0	ARA(x,y,z)	0.10	Mean(x)	0	ARA(x,y,z)	0.10
AAD(y)	0	Corr(x,z)	0.08	Kurt(x)	0	Corr(x,z)	0.07
Skew(x)	0	Mean(x)	0.08	Kurt(y)	0	Mean(x)	0
Kurt(x)	0	Kurt(x)	0	Skew(x)	0	Kurt(x)	0
Skew(y)	0	Skew(x)	0	Skew(y)	0	Skew(x)	0

The reduced subsets of features, before and after filtering, based on this algorithm are presented in Table 4.3. We find that CFS is particularly useful when data is filtered, for which it reduces the number of features from 19 to only 6, irrespective of whether activity S is considered or not.

4.3 Classification

Based on the success of other researchers in classifying a range of human activities using accelerometer data (see table 2), we chose the following five classifiers to evaluate and compare recognition performances for the indoor positioning activities considered in this paper:

- Decision Tree (DT): DT is a powerful and a popular tree-based tool for classification and prediction [27]. The classification process starts at the root of the tree and grows sequentially until reaching a leaf node. The central focus of the tree growing algorithm is testing and selecting the attribute with the most inhomogeneous class distribution, based on its information gain, explained in the feature reduction subsection in Section 4. A well-known algorithm, which has been widely used for building decision trees over the years, is C4.5 [30]. In this algorithm, Pruning is used to reduce the size of the tree to its optimal size, without reducing predictive accuracy. A tree that is too large risks overfitting the training data and poorly generalizing to new samples. A small tree might not capture important structural information about the sample space [22].

Table 4.3: Feature Reduction with Correlation Feature Selection (CFS)

E and L		E, L, and S	
Raw Data	Filtered Data	Raw Data	Filtered Data
Mean(y)	Mean(y)	Mean(z)	Mean(y)
Mean(z)	Std(z)	Std(y)	Std(z)
Std(x)	kurt(y)	Std(z)	Kurt(y)
Std(z)	AAD(x)	Skew(z)	AAD(x)
Skew(z)	Corr(x,y)	Kurt(z)	AAD(z)
Kurt(z)	Corr(y,z)	AAD(x)	Corr(y,z)
AAD(x)		AAD(z)	
Corr(x,y)		Corr(x,y)	
Corr(y,z)		Corr(y,z)	
ARA		ARA	

We note that the Decision Tree Pruning (DTP) process can be considered as a feature reduction, since it removes the redundant features. Therefore, we used the features appeared in the pruned trees as a reduced feature sets as shown in Table 4.4. Then we used these reduced feature sets as input to the five classifiers. To the best of our knowledge, no one has used features appeared in the pruned tree as a feature set to be used by other classifiers than DT. It is worth noting that the DTP reduced feature sets have the smallest number of features (see Table 4.4) compared to the resulted feature sets from the IG and CFS techniques (see Tables 4.2 and 4.3). DTP produces only 2 features in case of classifying E and L activities using the filtered data and 5 features for other scenarios.

- Decision Tables (DTL): DTL besides being one of the advanced rule-based classifiers that classify records using a collection of "If… Then…" rules, they are one of the simplest possible hypotheses spaces [19]. A decision table stores the input data in condensed form based on a selected set of attributes and uses it as a lookup table when making predictions [25]. Each entry in the table is associated with class probability estimates based on observed frequencies. The key to learning a decision table is to select a subset of highly discriminative attributes. The standard approach is to choose a set by maximizing cross-validated performance. Cross-validation is efficient for decision tables as the structure does not change when instances are added or deleted, only the class counts associated with the entries change.

- Naïve Bayes (NB): NB classifier employs a simplified version of Bayes formula [17], with strong (naïve) independent assumption, to decide which class a new instance belongs to. The posterior probability of each class is calculated, given the feature values present in the instance; the instance is assigned to the class with the highest probability. Equation (4.5) shows the NB classifier, which makes the assumption that feature values are

Table 4.4: Feature Reduction with Decision Tree Pruning (DTP)

E and L		E, L, and S	
Raw Data	Filtered Data	Raw Data	Filtered Data
Corr(y,z)	Std(x)	AAD(z)	Std(z)
Mean(y)	Std(z)	Corr(y,z)	AAD(z)
Mean(x)		Mean(y)	Std(x)
AAD(z)		Mean(x)	Kurt(y)
Corr(x,z)		Corr(x,z)	Mean(x)

Note: When the data were filtered, no correlation features appeared in the reduced set.

statistically independent within each class.

$$c_{NI} = arg\, max_{c_i \in C} P(c_i) \prod_j P(f_j|c_i) \tag{4.5}$$

Where c_{NI} is the class of the new instance, $C = (c_1, c_2, \ldots, c_n)$ is the classes, and f_j is the feature value.

- K-Nearest Neighbour (KNN): KNN is one of the simplest machine learning algorithms. It is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification [8]. The KNN algorithm classifies new examples based on identifying the k-nearest example(s) in the training set of features according to some distance metric (In WEKA, Euclidean distance is used) and then taking the majority vote. The parameter k is a positive integer, typically small. If k=1, then the object is simply assigned the class of its nearest neighbour. The best choice of k depends upon the data.

- Multilayer Perceptron (MLP): MLP represents the most prominent and well researched class of ANNs in classification, implementing a feedforward and supervised paradigm [26]. Although many types of neural networks can be used for classification purposes, MLP remains one of the fastest tool for neural networks studies [29] [33]. MLP consists of several layers of nodes, interconnected through weighted acyclic arcs from each preceding layer to the following, without lateral or feedback connections. Each node calculates a transformed weighted linear combination of its inputs.

The following section presents the results of applying the five classifiers to the different feature sets including the original feature set and the reduced feature sets based on the three techniques, IG, CFS, and DTP.

5 Results

To train and test the classifiers with different data sets, we apply a k-fold cross-validation scheme [18]. The entire data set is divided in to k sets, where k-1 of them are used for training and one set for testing. This is repeated k

Table 5.1: ACCURACIES (%) of Classifying only E and L activities

Data	Feature Set	DTL	DT	NB	KNN	MLP
Raw	Original (19)	81.05	82.42	86.23	**90.39**	87.56
	IG (11)	80.56	80.69	**88.25**	86.60	84.98
	CFS (10)	80.21	81.60	**90.22**	85.71	84.61
	DTP (5)	83.21	86.65	88.65	**89.28**	86.32
Filtered	Original (19)	93.21	**97.80**	94.01	94.76	95.70
	IG (17)	93.29	**97.80**	94.02	93.57	94.62
	CFS (6)	93.40	**96.61**	95.44	94.43	96.08
	DTP (2)	93.91	**97.80**	93.69	96.88	96.88

times and then the average of the results is reported. We used k=10. We first present HAR accuracy results when recognition is performed with only E and L activities, followed by the case when S is introduced. Finally, we investigate how the sampling window size affects HAR performance.

5.1 Classifying only E and L activities

Table 5.1 summarizes the results of the recognition accuracies, based on the raw and filtered data, of the five used classifiers. The feature set column shows the different feature sets including the *original* set of 19 features and the reduced feature sets from IG, CFS, and DTP techniques (number of features shown in the parenthesis). For each feature set, the highest accuracy obtained is shown in bold. We make the following observations:

- Using the raw data (no filtering), the highest accuracy that can be achieved is 90%, either with the original feature set fed to a KNN classifier, or with the reduced set obtained from CFS in conjunction with a NB classifier. The smallest feature set (DT Pruning) achieves 89% accuracy with the KNN classifier.

- A considerable improvement in accuracies (ranging between 8% and 15%) is achieved when the data are filtered with a moving average filter. In fact an accuracy as high as 97.80 % can be reached, with only 2 features resulting from DTP reduction technique and a DT classifier.

- For filtered data, DT outperforms all other classifiers irrespective of the feature set used.

- Another interesting and useful finding lies in the difference between the reduced feature sets obtained with DT Pruning for raw and filtered data (compare the first and second columns in Table 4.4). Filtering removes any *correlation features* from the feature set. The significance of this observation comes from the fact that a correlation feature needs to process entries from two dimensions potentially increasing the feature computation overhead. Therefore, we can expect that filtering overhead would be compensated by the savings in feature computation. Thus, the noticeable increase in recognition accuracy due to filtering could come without any net increase in computation overhead.

Table 5.2: ACCURACIES (%) of classifying E, L, and S activities

Data	Feature Set	DTL	DT	NB	KNN	MLP
Raw	Original (19)	80.59	82.97	87.49	**89.20**	87.86
	IG (14)	79.58	81.44	**88.05**	85.46	86.34
	CFS (10)	80.25	82.01	**88.32**	87.44	84.34
	DTP (5)	83.06	87.95	**89.46**	87.95	88.74
Filtered	Original (19)	80.22	90.57	85.84	85.67	**91.40**
	IG (16)	80.40	90.57	84.37	82.18	**91.11**
	CFS (6)	82.23	89.66	89.56	84.88	**93.01**
	DTP (5)	82.38	90.63	88.32	93.77	**94.42**

Table 5.3: Confusion Matrix of using the MLP classifier coupled with the DTP reduced feature set on the filtered data

		Classified as		
		E	L	S
Actual Activity	E	61	0	3
	L	4	74	2
	S	2	2	29

5.2 Classifying E, L and S activities

To assess the impact of the presence of S activity on the recognition accuracy, we have recomputed the feature reductions and classifications with the three activities, E, L, and S (see Table 5.2). We make the following observations:

- Inclusion of S has no noticeable effect when raw data is used for HAR. This could be due to the noise content in the raw data, therefore, the noise introduced by S is not noticeable.
- For filtered data, the presence of the activity S resulted in a noticeable reduction in classification accuracies. It is clear from Tables 5.1 and 5.2, that the presence of activity S has a two-fold effect. The maximum accuracy of 97.80% obtained from DTP has dropped to 94% while the number of featues has increased from only 2 to 5.
- For filtered data, MLP classifier outperforms all other classifiers irrespective of the feature set used.
- Finally, using the confusion matrix (see Table 5.3), we investigate how the S activity confuses recognition of E and L activities. We find that there is a 12% probability (4 out of 33) that S activity may be incorrectly recognized as E or L.

5.3 Effect of Sampling Window Size

The results discussed in the preceding sections are all based on a sampling window size of 100 samples, which corresponds to 5 seconds worth of acceleration data at 20Hz. As explained earlier, our decision to choose a 5 sec window was based on earlier research [9] which found that we need at least 5 sec to detect human activity. We investigate the validity of this finding by recomputing our

Table 5.4: The effect of using Different Window Length

Window	E and L		E, L, and S	
	Features No.	Accuracy (%)(DT)	Features No.	Accuracy (%) (MLP)
5 sec	2	97.80	5	94.42
4 sec	4	94.58	5	94.93
3 sec	3	95.33	6	94.74
2 sec	4	92.87	10	92.94
1 sec	7	85.96	9	89.60

results for several smaller window sizes (see Table 5.4) using the DTP reduced feature set of the filtered data and the classifiers that achieved the highest accuracies (DT in case of Classifying only E and L activities and MLP in case of classifying E,L and S). Indeed, we find that smaller windows generally reduce achievable accuracy, and the number of features needed for classification increases as well. For example, if we reduced the window size from 5 sec to 1 sec, we would not only reduce the accuracy from 97% to 85%, but also increase the number of features from 2 to 7 (for the classification of E and L only). Increasing accuracy and reducing number of features for small windows, therefore, would be an interesting and challenging future work.

6 Conclusions and Future Work

With access to indoor maps, HAR becomes a potential tool for indoor positioning. E and L are two widely performed indoor activities with clear positional reference to most indoor maps, hence have high potential for indoor positioning. Unfortunately, for both E and L, the user typically stands still, making their recognition a challenging problem. In this paper, we have investigated the possibility of a smartphone accurately detecting these two activities with minimal sensor and HAR overhead. Based on real experimental data, we have shown that it is possible to recognize E and L with 97% accuracy using only accelerometer signals and minimal HAR computations. Our conjecture is that this high accuracy is due to the differences in the ways an escalator and a lift vibrates, which is rooted to the design and riding quality standards for such equipment. We have, however, found that another common activity, S, which has no contribution to indoor positioning, may significantly confuse the recognition of E and L.

A natural continuation of the current work would be to consider additional activities that also have high potential for indoor positioning. One such activity would be stair climbing with its potential confusion with the walking activity. An interesting future work would be to actually implement a working prototype on a smartphone and measure its performance with live experiments, as well as quantify the power consumption of the HAR algorithms. It will be a continuous effort to further improve the recognition accuracy and the power consumption of the HAR techniques. Finally, it would be a challenging but useful exercise to achieve good accuracy with smaller sampling windows.

Bibliography

[1] The application used to collect the accelerometer reading. http://androidapplications.com/23458-accelerometervalues. Accessed on 1 August, 2012.

[2] Centro Bankstown. http://www.centrobankstown.com.au. Accessed on 22 Septemper, 2012.

[3] Elevator and Escalator Ride Quality Testing ISO 18738. http://www.onsiteacoustictesting.com/?p=284. Accessed on 24 Septemper, 2012.

[4] NICTA ATP. http://www.woodhead.com.au/projects/nicta-australian-technology-park-sydney-new-south-wales/. Accessed on 22 Septemper, 2012.

[5] Parramatta Westfield. http://www.westfield.com.au/parramatta. Accessed on 22 Septemper, 2012.

[6] Sydney Airport Map. http://www.sydneyairport.com.au/find/airport-maps.aspx. Accessed on 15 August, 2012.

[7] WEKA Software. http://www.cs.waikato.ac.nz/ml/weka/. Accessed on 15 August, 2012.

[8] David W. Aha, Dennis Kibler, and Marc K. Albert. Instance-based learning algorithms. *Machine Learning*, 6(1):37–66, 1991.

[9] Kerem Altun, Billur Barshan, and Orkun Tunçel. Comparative study on classifying human activities with miniature inertial and magnetic sensors. *Pattern Recognition*, 43(10):3605–3620, 2010.

[10] Mohamed Amir. Energy-aware location provider for the android. Master's thesis, Computer and systems engineering department, Alexandria university, Egypt, 2010.

[11] M.A. Ayu, T. Mantoro, A.F.A. Matin, and S.S.O. Basamh. Recognizing user activity based on accelerometer data from a mobile phone. In *Proceedings of the IEEE International Symposium on Computers and Informatics (ISCI 2011)*, Kuala Lumpur, Malaysia, 20-23 March 2011.

[12] Jonghun Baek, Geehyuk Lee, Wonbae Park, and Byoung-Ju Yun. Accelerometer signal processing for user activity detection. In *Knowledge-Based Intelligent Information and Engineering Systems*, volume 3215 of *Lecture notes in Computer Science*, pages 610–617. Springer Berlin / Heidelberg, 2004.

[13] L. Bao and S. S. Intille. Activity recognition from user-annotated acceleration data. *Pervasive Computing*, 3001:1–17, 2004.

[14] Mathieu Cunche, Mohamed Ali Kaafar, and Roksana Boreli. I know who you will meet this evening! linking wireless devices using wi-fi probe requests. In *13th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, San Francisco, California, 25–28 June 2012.

[15] D. Gusenbauer, C. Isert, and J. Krösche. Self-contained indoor positioning on off-the-shelf mobile devices. In *"International Conference on Indoor Positioning and Indoor Navigation (IPIN10)*, Zürich, Switzerland, 15–17 Septemper 2010.

[16] Mark A. Hall. *Correlation-based feature selection for machine learning.* PhD thesis, Department of Computer Science, University of Waikato, Hamilton, New Zealand, 1998.

[17] George John and Pat Langley. Estimating continuous distributions in bayesian classifiers. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann, 1995.

[18] Satyen Kale, Ravi Kumar, and Sergei Vassilvitskii. Cross-validation and mean-square stability. In *Proceedings of the 25th International Conference on Supercomputing*, Loews Ventana Canyon Resort, Tucson, Arizona, 31 May - 4 June 2011.

[19] Ron Kohavi. The power of decision tables. In *Proceedings of the European Conference on Machine Learning*. Springer Verlag, 1995.

[20] Jennifer R. Kwapisz, Gary M. Weiss, and Samuel A. Moore. Activity recognition using cell phone accelerometers. *ACM SIGKDD Explorations Newsletter*, 12(2):74–82, 2010.

[21] GP Lorsbach. Analysis of elevator ride quality, vibration. *Elevator World*, 51(6):108–113, 2003.

[22] Yishay Mansour. Pessimistic decision tree pruning based on tree size. In *Proceedings of the Fourteenth International Conference on Machine Learning*. Morgan Kaufmann, 1997.

[23] T. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997.

[24] R. Mortier, C. Greenhalgh, D. McAuley, A. Spence, A. Madhavapeddy, J. Crowcroft, and S. Hand. Personal containers or, your life in bits. In *Digital Futures Workshop*, Nottingham, UK, 11 October 2010.

[25] Srinivas Murti and Mahantappa. Using rule based classifiers for the predictive analysis of breast cancer recurrence. *Journal of Information Engineering and Applications*, 2(2):12–19, 2012.

[26] S.K. Pal and S. Mitra. Multilayer perceptron, fuzzy sets, and classification. *IEEE Transactions on Neural Networks*, 3(5):683 –697, 1992.

[27] J. Parkka, M. Ermes, P. Korpipaa, J. Mantyjarvi, J. Peltola, and I. Korhonen. Activity classification using realistic data from wearable sensors. *Information Technology in Biomedicine, IEEE Transactions on*, 10(1):119–128, 2006.

[28] Avinash Parnandi, Ken Le, Pradeep Vaghela, Aalaya Kolli, Karthik Dantu, Sameera Poduri, and Gaurav S. Sukhatme. Coarse in-building localization with smartphones. In *Mobile Computing, Applications, and Services*, volume 35 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 343–354. Springer Berlin / Heidelberg, 2010.

[29] Lutz Prechelt. A quantitative study of experimental evaluations of neural network learning algorithms: Current research practice. *Neural Networks*, 9(3):457 – 462, 1996.

[30] J. Ross Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc, San Francisco, CA, USA, 1993.

[31] Nishkam Ravi, Nikhil Dandekar, Preetham Mysore, and Michael L.l Littman. Activity recognition from accelerometer data. In *Proceedings of the 17th conference on Innovative applications of artificial intelligence*, Pittsburgh, Pennsylvania, 9-13 July 2005.

[32] Steven W. Smith. *The Scientist and Engineer's Guide to Digital Signal Processing*, chapter 15. California Technical Publishing, San Diego, California, 1999.

[33] Mevlut Ture, Imran Kurt, A. Turhan Kurum, and Kazim Ozdamar. Comparing classification techniques for predicting essential hypertension. *Expert Systems with Applications*, 29(3):583 – 588, 2005.

[34] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann Publishers Inc, 1999.

[35] Zhaohui Zheng. Feature selection for text categorization on imbalanced data. *ACM SIGKDD Explorations Newsletter*, 6(1):80–89, 2004.

[36] Wei zhong Wang, Yan wei Guo, Bang yu Huang, Guo ru Zhao, Bo qiang Liu, and Lei Wang. Analysis of filtering methods for 3d acceleration signals in body sensor network. In *Proceedings of the IEEE International Symposium on Bioelectronics and Bioinformatics (ISBB 2011)*, Suzhou, China, 3-5 November 2011.

[37] Zhenyun Zhuang, Kyu-Han Kim, and Jatinder Pal Singh. Improving energy efficiency of location sensing on smartphones. In *Proceedings of the 8th international conference on Mobile systems, applications, and services (MobiSys '10)*, San Francisco, California, 15–18 June 2010.